DARIA: Designing Actuators to Resist Arbitrary Attacks Against Cyber-Physical Systems

Jairo Giraldo
Electrical and Computer Engineering
University of Utah

Sahand Hadizadeh Kafash,
Justin Ruths
Mechanical Engineering
University of Texas at Dallas

Alvaro A. Cardenas
Computer Science and Engineering
University of California, Santa Cruz

***Abstract*—In the past decade we have seen an active research community proposing attacks and defenses to Cyber-Physical Systems (CPS). Most of these attacks and defenses have been heuristic in nature, limiting the attacker to a set of predefined operations, and proposing defenses with unclear security guarantees. In this paper, we propose a generic adversary model that can capture any type of attack (our attacker is not constrained to follow specific attacks such as replay, delay, or bias) and use it to design security mechanisms with provable security guarantees. In particular, we propose a new secure design paradigm we call DARIA: Designing Actuators to Resist arbItrary Attacks. The main idea behind DARIA is the design of physical limits to actuators in order to prevent attackers from arbitrarily manipulating the system, irrespective of their point of attack (sensors or actuators) or the specific attack algorithm (bias, replay, delays, etc.). As far as we are aware, we are the first research team to propose the design of physical limits to actuators in a control loop in order to keep the system secure against attacks. We demonstrate the generality of our proposal on simulations of vehicular platooning and industrial processes.**

***Index Terms*—Cyber-Physical Systems, Optimal Defense, Security-by-Design.**

1. Introduction

Securing computing systems that interact and change the physical world is becoming a priority as cars, drones, control systems, and medical devices become more connected and controlled by software. In the past decades there have been several confirmed attacks to control systems, including attacks to a sewage control system in Australia [1], a nuclear enrichment facility in Iran [2], the power grid of Ukraine [3], a steel mill in Germany [4], a paper mill in Louisiana [5], and an unidentified industrial control system in the Middle-East [6]. In all these cases an attacker was able to partially compromise a system and then launch control signals that drove these systems to cause accidents and damage (e.g., the attacker of the sewage system in Australia caused more than 750,000 gallons of untreated sewage water to be released into parks, rivers, and hotel grounds causing loss of marine life, jeopardizing public health, and costing more than $200,000 in cleanup and monitoring costs).

In this paper we consider this threat model, where an attacker has already partially compromised a system (having access to a controller, sensor, or actuator) and tries to drive the system to an unsafe state. While the research community has been active in trying to detect and prevent these attacks, we have found three major limitations of previous work.

First, most of the threat and attack models in previous work presented in security conferences assume that the control signal of the attacker is specified a priori and constrained to a few parametric models. For example, a scaling attack [7] takes a compromised value u_t (at time t) and scales it with a constant value (γ) to produce the attack $a_t = \gamma u_t$, a bias attack [8], [9] takes a compromised value u and adds a constant bias, such as *decrease the water level sensor by 1mm each second* [8], producing the attack time series $a_t = u_t - b$. Abrupt-attacks take the maximum possible value a signal can have [8]–[10] (e.g., set a sensor to the highest level) [8], thus producing the attack signal $a_t = \max u$, delay attacks take a compromised signal u_t and delay it in time, giving the attack time series $a_t = u_{t-\tau}$ [7], and random attacks replace the compromised signal with an attack a_t chosen from a fixed random probability distribution [11], [12]. While all of the examples presented so far are from cyber-security conferences, the literature in control systems has very similar attacker models with replay attacks [13], [14] ($a_t = u_{t-\tau}$), or scaling attacks [15] ($a_t = Tu_t$, with T a matrix). In contrast, in this paper we consider that the attacker has full control of the attack signals, and is not constrained to a parameterized attack; in particular, at every single time t, the attacker can chose any arbitrary value $a_t \in \mathcal{A}$, where $\mathcal{A}$ is the space of all possible physical values a_t can take. Letting the attacker select any arbitrary value at any time is important because if the defender constrains the attacker to follow a small subset of parametric attacks like previous work, then defenders cannot guarantee that their security solution will be valid for all possible attack strategies.

Second, most of the previous work dealing with partially compromised systems has focused on detecting attacks [8]–[10], [16]–[18]. While there are several efforts trying to protect these systems from attacks such as securing in-vehicle car networks [19], [20], the problem of how the physical dynamics of a control system evolve when the system is under attack remains largely unaddressed. Perhaps the closest work to our proposed effort is the use of resilient estimation in control systems, such as adaptive cruise control [12], [21], where the idea is to detect an estimation inconsistency and discard these

DOI 10.1109/EuroSP48549.2020.00029

measurements while predicting the future based on the last known measurement. However, as shown by Urbina et al. [16], there will always be attacks that can go undetected if the attacker chooses the attack time series to follow closely the expected behavior of the system. In addition, these resilient estimators were evaluated with parametric attacks such as a delay attack, or a denial-of-service attack, and again, do not consider a powerful adversary that can select an arbitrary value that avoids detection. Finally, resilient estimation approaches work only when the adversary compromises sensor data, but if the adversary compromises a control algorithm, then the estimation algorithm will work correctly and it won't prevent the adversary to drive the system to an unsafe space.

Third, most previous efforts show specific examples where under a set of example attacks, the system is able to survive them. However, they do not offer formal proofs of security that guarantee that no matter what strategy the attacker selects (e.g., attacks outside of the examples shown), the system will remain under safe operation and will survive the attack.

In this paper we address these three limitations by proposing a new design paradigm for securing control systems against arbitrary attackers. In particular we propose DARIA: Designing Actuators to Resist ArbItrary Attacks and show how it provides **provable security guarantees**. We show that by properly designing the limits of an actuator we can prevent the attacker from driving the system to unsafe regions, no matter what strategy (bias, delay, etc.) or point of entry (sensor, actuator, controller) the attacker selects.

The basic idea behind our insight can be easily explained by an example: assume there is a valve with operating range from 0 to 470 liters per second that lets water into a tank. If an attacker compromises the control signal and can use this valve to overflow the tank, then we know the system is not safe. However, if by installing a different valve with operating range from 0 to 330 liters per second the adversary cannot overflow the tank, then we can say that with the second valve the system is secure. We take this intuitive idea and formalize it to propose a rigorous design algorithm that can be proven secure against a powerful adversary. As far as we are aware, we are the first research team to propose the design of physical limits to actuators in a control loop, in order to keep the system secure against attacks. Our solution can be implemented physically (in the actuator itself) or in software.

In particular, we design an efficient algorithm that gives an outer bound to the set of possible reachable states by the adversary, and then, show how to design the bounds in the actuators of a system to prevent the adversary from reaching unsafe states.This procedure only needs to be done at the design state of the CPS, and will prevent any adversary which compromises the system at any future point, from violating the safety limits of the system. In addition, it does not matter if the adversary compromises the sensor, actuator, or controller, as our system is designed such that no actuation can drive the system to unsafe regions.

The rest of the paper is organized as follows: In Section 2 we introduce the main problem, the adversary model, and the definition of security. We present the security analysis and the mechanism to design safe CPS in Section 3. In Section 4 we validate our approach in two case-studies, a classical process control problem called the quadruple tank process, and a vehicular platooning scenario. We finalize the paper with discussions on Limitations, Future Work, and Related work.

2. Problem Formulation

Physical processes can be represented by a set of differential (or difference) equations that capture their dynamic behavior. One of the most popular ways to model these systems are linear time invariant systems [16]:

$$x_{t+1} = Ax_t + Bu_t \tag{1}$$

where at each t^{th} time sample, $x_t \in \mathcal{X}$ is the vector of size n that represents the system states (e.g., temperature, velocity, pressure, etc.), $u_t \in \mathcal{U}$ is the vector of size m that represents the control inputs (e.g., valve position, acceleration, steering angle, etc.). Matrices $A \in \mathbb{R}^{n\times n}$, and $B \in \mathbb{R}^{n\times m}$ indicate how the current states and control action will affect the future states.

Most of the literature in control systems assumes $x_t \in \mathbb{R}^n$ and $u_t \in \mathbb{R}^m$; that is, they assume that the states x and the control (and ultimately actuation) commands u are unbounded and could take near infinite values [13], [14], [22]. While this assumption is clearly impossible in real life—a water valve cannot allow an infinite amount of water to flow into a tank—the reason for this unbounded assumption is twofold: First, allowing x and u to be unbounded allows the simplification of the mathematical analysis of the system and can allow researchers to obtain clean mathematical expressions for the control algorithm and for the behavior of the system; this assumption has allowed proofs related to system controllability and observability [23] or the optimal estimation of states in a stochastic system with a Kalman filter [24]. Second, for most practical systems this assumption makes sense, as Equation (1) usually represents the behavior of a system close to their operational point, and the goal is to keep the states x_t close to the selected operational points, which are selected to be 0 for practical purposes—therefore, popular control algorithms (such as PID control) which create control inputs u_t proportional to x_t will therefore generate control values that remain small and bounded for most practical scenarios.

While the general assumptions on the behavior of control systems are satisfied for most practical use-cases and, therefore, they can be analyzed (without an adversary) using unbounded states and unbounded control signals, the existence of an adversary will break these assumptions.

2.1. New Adversary Model

In her National Computer Systems Security Award Acceptance Speech in 1999, Dorothy Denning explained succinctly how systems that are proven secure, are proven under a specific adversary assumptions, and then gave examples of how the way to crack a system is to step outside the box, to break the rules under which the protocol was proved secure [25]. Over the years, system after system has been defeated by adversaries that break the assumptions of the model. The most recent high-profile example is the case of attacks against key handshakes in WPA2 [26] which were proven secure [27] under a model that did not capture key installation. In an effort to increase the confidence of a security proof, adversary models used in formal proofs tend to be as general as possible [28]–[30], e.g., by assuming adversaries to be any polynomial time algorithm [29], [30] (without parameterizing the specific attacker algorithm used to crack the system).

In this paper we attempt to bring a similar generic adversary model so that we capture a powerful adversary that has compromised a control signal and which will have the ability to change it arbitrarily. Our attacker is not constrained to follow specific attacks such as replay, delay, scaling or bias attacks previously considered in the literature. Using this general adversary model, we now define what it means to be secure.

2.2. Definition of Security

Intuitively, a CPS is secure if there exists no attack that can drive the system to an unsafe (dangerous) state. We now define this statement formally:

Dangerous States. The set of dangerous states $\mathcal{D}$ can be represented mathematically as the union of κ half spaces of the form $c_j^\top x \geq b_j$, as follows

$$\mathcal{D} := \left\{ x \in \mathbb{R}^n \;\middle|\; \bigcup_{i=1}^{\kappa} c_i^T x \geq b_i \right\}. \tag{2}$$

For example, if our states are $x_1 = temperature$ and $x_2 = pressure$, a dangerous state could be when the temperature is greater than $100^o\ F$ and pressure is greater than $50\ Pa$. Therefore, we have two half-spaces, one with $c_1 = [1,0]^\top, b_1 = 100$ and the other with $c_2 = [0,1]^\top, b_2 = 50$.

Let a_t denote the control signals that can be manipulated by the attacker (whether this is indirect with a sensor compromise, or directly by attacking the actuator or the control signal), then the CPS under attack has the following dynamics:

$$x_{t+1} = Ax_t + Ba_t \tag{3}$$

Definition 1. **Secure CPS**: We say that the CPS in Equation (3) is **secure** *if and only if* $\neg\exists(a_0, a_1, \ldots a_T)$ for any arbitrary time duration T that satisfies the following proposition: $\exists t*$ such that $x_{t*} \in \mathcal{D}$ and $\forall t \in \{0, \ldots, T\} a_t \in \mathcal{U}$.

The definition of security above states that there is no attack sequence that causes the state of the system (at any point in time) to reach the unsafe states $\mathcal{D}$. The attack sequence has to be in the bounded space of possible actuation commands $\mathcal{U}$, which depends on the physical limits of the actuators—e.g., a valve can only allow water at a certain flow rate, or vehicles can accelerate at a maximum of (for example) $9.8m/s^2$. The attacker cannot exceed these values.

We will show how to prove that a CPS is secure in Section 3. In particular in the next section we will introduce a *sufficient condition* to prove that a given CPS is secure. If we cannot prove security of the system with the current conditions, we then show that if there exists a set $\mathcal{U}$ that will pass the sufficient condition, then we can find it (and therefore we can prove that we can make the CPS secure).

We will also show in the examples of the paper that the above definition of security also works when the operation of the system is given by the following equation:

$$x_{t+1} = Ax_t + B_1 u_t + B_2 a_t \tag{4}$$

where $B_1 u_t$ denotes the part of the system that is still controlled by the defender, and $B_2 a_t$ the part of the system compromised by the attacker.

In the next section we will describe how our approach solves the following two problems:

- *Counterexamples:* If the current configuration of the CPS system is insecure, our attack design attempts to find a **feasible** attack sequence $a_0, a_1, \ldots, u_T$ that will drive the system to an unsafe state at time T. We will show how this approach can find attacks that are not intuitive, and which would be very difficult for an attacker to find without our formulation. One such example will be presented in Section 4.2 where we show that the feasible attacks require the attackers to create an oscillation in the system.
- *Security:* We show how to efficiently find an approximation to the outer bounds of all possible states that the attacker can drive the system to, and then show how adjusting the bounds on the control signal space $\mathcal{U}$, we can iteratively find an secure configuration of the CPS system.

3. Security Analysis and Design of Safe CPS

In order to test if a system represented by equation (3) is secure according to Definition 1, we study the evolution of the system after t time steps. To this end, we can exploit the recursive structure of Equation (3) as follows:

$$\begin{aligned} x_1 &= Ax_0 + Ba_0 \\ x_2 &= Ax_1 + Ba_1 = A(Ax_0 + Ba_0) + Ba_1 \\ &\vdots \\ x_t &= A^t x_{t-1} + Ba_{t-1} = A^t x_0 + \sum_{j=0}^{t-1} A^{t-1-j} Ba_j. \end{aligned}$$

Thus, for any initial state x_0, and the input sequence $a_0, a_1, \ldots, a_{t-1}$, the system states at any time t are:

$$x_t = Ax_0 + \sum_{j=0}^{t-1} A^{t-1-j} Ba_j. \tag{5}$$

The second term in the right can be rewritten as

$$[A^{t-1}B, A^{t-2}B, \ldots, A^0 B] \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_{t-1} \end{bmatrix}$$

Let $H = [A^{t-1}B, A^{t-2}B, \ldots, A^0 B]$ and $\boldsymbol{U} = [a_0, \ldots, a_{t-1}]^\top$ be the complete input sequence up to time $t-1$. We can rewrite (5) as follows:

$$x_t = A^t x_0 + H\boldsymbol{U}. \tag{6}$$

The above equation gives us the expected trajectory of a system under attack. We now use that equation to attempt to find feasible attacks.

3.1. Feasible Attacks

In this section we discuss our first step in the security analysis of a CPS. In particular we show how the attacker can attempt to find a feasible attack; that is, an attack signal that drives the system to one of the unsafe regions. We will show how our problem formulation can find attacks that are not intuitive. One such example will be presented in Section 4.2 where we show that the feasible attacks require the attackers to create an oscillation in the system.

We assume the adversary wants to drive the system states (e.g., temperature, velocity, pressure) to a desired unsafe state $x_d \in \mathcal{D}$ in at most $\mathcal{T}$ units of time. If there is an attack vector U in Equation (6) that satisfies $x^d = A^{\mathcal{T}} x_0 + HU$ and that each u_t is bounded by $\mathcal{U}$ then we know the system is not secure.

The question is how can we find this attack in a large search space. One of our insights is that we can formulate this search problem (and all the other search problems presented in this paper) as a semidefinite programming problem [31]. Semidefinite programming is a relatively new field that is growing in popularity because of the various applications in operations research and engineering. In semidefinite programming we search for a feasible solution subject to the constraint that an affine combination of symmetric matrices is positive semidefinite. Such a constraint is nonlinear and nonsmooth, but convex, so positive definite programs are convex optimization problems can be solved with primal-dual interior-point methods [32].

In addition to performing a search to find a feasible solution U, semidefinite programming allows us to define an optional objective function (like most optimization approaches) in case we want to find not only a feasible solution, but the best feasible solution according to the objective function. In general, finding feasible attacks for a single state x^d can be formulated as the following optimization problem with objective function 0 (i.e., we do not care about optimizing any objective, although that can be changed to include an objective function).

Attack Discovery

$$\begin{aligned}
&\min_{U} 0 \\
&s.t. \\
&x_d = A^{\mathcal{T}} x_0 + HU \qquad\qquad (7)\\
&u_j \in \mathcal{U}, \quad \text{for all } j = 0, 2, \ldots, \mathcal{T} - 1 \\
&x_T = x_d
\end{aligned}$$

In order to solve this problem we use YALMIP [33], a tool to formulate and solve complex optimization problems interfacing many external commercial and non-commercial solvers. The particular solver we use in this work is SDPT3 [34], which uses primal-dual path following algorithms to solve semidefinite linear and quadratic optimization problems. The idea behind this algorithm is that at each iteration it tries to decrease the duality gap as much as possible while keeping the iterates close to the central path, which ensures that the algorithm converges to a solution [35]. Another reason we chose SDPT3 over other solvers is for its ability to solve a wide variety of convex optimization problems, including the ones with objective function of the form $-\log\det P$ (which we will use in Section 3.2) which can be solved more accurately than with other solvers (e.g., the SEDUMI solver is unable to exactly compute $-\log\det P$ so it uses a linearized approximate expression), and the ability to solve Linear Matrix Inequalities, which we will use in the next section.

The characteristic of our proposed attacker model is that they can generate *any* allowable attack sequence by compromising directly the controller (gaining access to the PLC or intercepting the control commands) or the sensors. The only limitation of our attacker is given by the physics of the actuation devices, e.g., min/max voltage, min/max acceleration. This means that this is the strongest type of attack.

While our problem formulation and its solution via SDPT3 is sound, in the sense that any feasible solution will show an attack that drives the system to an unsafe space, our formulation is not complete. If we cannot find an attack, this does not necessarily mean that the system is secure. In the next section we solve this problem by presenting a condition that can guarantee that a system is provably secure (i.e., attacks that drive the system to an unsafe space do not exist).

In particular, in the next subsection we will introduce new tools to help us find approximations of the reachable set, such that it is possible to determine if a system is secure by solving an efficient search problem. If the system is not secure, we will redesign it in order to guarantee that it remains safe for any input sequence.

3.2. Security-by-Design

Our goal is now twofold: (1) to find conditions that give a sufficient guarantee for security (no attack can damage the system) and (2) to redesign the actuation constraints in order to guarantee that no unsafe state is feasible (i.e., the unsafe set does not overlap with the reachable set). We start by finding sufficient conditions for the nonexistence of attacks.

Definition 2. For a given initial state x_0, the reachable set $\mathcal{R}$ corresponds to the set of states $x \in \mathcal{X}$ that can be reached starting from x_0 by *any arbitrary sequence* $u_0, u_1, \ldots,$ where each input is bounded by $u_{min} \leq u_{i,t} \leq u_{max}$.

Notice that in Definition 2, the reachable set includes the states that can be reached by infinite input sequences. As a consequence, obtaining the exact reachable set for a given system is computationally intractable; however, there are tools from control theory to find ellipsoids that contain the reachable set. The ellipsoid equation can be represented using matrix and vectors operations. For example, suppose we want to represent an ellipsoid in 2 dimensions, x_1, x_2. A typical equation of an ellipse centered in zero and rotated 45^o is $x_1^2 - x_1 x_2 + x_2^2 = 1$. The matrix representation is then $x^\top P x = 1$, with $P = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}$. Similarly, we can represent any higher dimension ellipsoid by just defining an adequate matrix P. Our goal is to find an ellipsoid which encapsulates the entire reachable set. This is what will give us the **sufficient** condition for security: if our ellipsoid does not intersect with the unsafe region, then we know that there is not possible trajectory that can reach the unsafe region.

We define these ellipsoids as:

$$\mathcal{E}(P) := \left\{x \in \mathbb{R}^n \mid x^T P x \leq 1\right\}, \tag{8}$$

where $P \in \mathbb{R}^{n \times n}$ is a positive-definite matrix.

We can define $d_{\mathcal{E}} \in \mathbb{R}$ as the signed distance between the ellipsoid $\mathcal{E}(P)$ and the half spaces of the form $c_i x \geq b_i$ for $i = 1, \ldots, \kappa$ that represent the dangerous states as described in Section 2.2. Therefore, according to [36], we have that

$$d_{\mathcal{E}} = \min_i \left(\frac{|b_i| - \sqrt{c_i^\top P^{-1} c_i}}{\sqrt{c_i^\top c_i}} \right). \tag{9}$$

If $d_{\mathcal{E}} \leq 0$, then the ellipsoid overlaps with at least one half space. On the other hand, $d_{\mathcal{E}} > 0$ implies that there is no overlap.

If there exists at least one element of the dangerous states $\mathcal{D}$ that is also contained in the reachable set $\mathcal{R}$, the system is not secure. **Therefore, if we are able to find an ellipsoid that does not overlap with the dangerous states (i.e., $d_{\mathcal{E}} > 0$), since $\mathcal{R} \subseteq \mathcal{E}(P)$, we can guarantee that the reachable set does not overlap with the dangerous states either, and the system is secure**. We will introduce the mathematical formulation and the optimization problem but the details of the proofs can be found in Appendix A.

Before introducing our main result, we can concisely write each actuator bound as $|u_{i,t}|^2 \leq \gamma_i$ for $i = 1, \ldots, m$. To ensure the convexity of the optimization problem, the entire set of actuator bounds $\mathcal{U}$ can be approximated to an ellipsoid of the form $\widetilde{\mathcal{U}} = \{u \in \mathbb{R}^m : u^\top R u \leq m\}$, where $R = \operatorname{diag}\left(\frac{1}{\gamma_1}, \ldots, \frac{1}{\gamma_m}\right)$, such that $\mathcal{U} \subseteq \widetilde{\mathcal{U}}$, where the notation diag(a) for a vector $a \in \mathbb{R}^n$ refers to a $n \times n$ diagonal matrix with the elements of a in its diagonal.

Now, the following Proposition allows us to compute an ellipsoid that contains the reachable set for given bounds R.

Proposition 1 (Reachable Set Approximation). For the LTI system (1) with controllable pair (A, B), and upper bounds $\gamma_i \geq 0$, $i = 1, \ldots, m$ collected in R, if there exists an $a \in (0, 1)$ for which the positive definite matrix P is a solution of the following convex optimization problem:

$$\begin{cases} \min\limits_{P} \; -\log\det P, \\ \text{s.t. } P > 0, \text{ and} \\ \begin{bmatrix} aP - A^T P A & -A^T P B \\ -B^T P A & \frac{1-a}{m} R - B^T P B \end{bmatrix} \geq 0, \end{cases} \tag{10}$$

then $\mathcal{R} \subseteq \mathcal{E}(P)$ and $\mathcal{E}(P)$ has minimum volume.

The objective $-\log\det P$ tries to minimize the volume of the ellipsoid such that it is as close as possible to the reachable set. Now, since our intent is to redesign the bounds R to avoid dangerous states, we can formulate an optimization problem incorporating the dangerous states $\mathcal{D}$ according to the following Theorem:

Theorem 1 (Bound Design). Consider the LTI system (1) with controllable pair (A, B) and a set of dangerous states $\mathcal{D}$ defined by (2). If there exists an $a \in (0, 1)$ for which the positive definite matrix P is a solution of the following convex optimization problem:

$$\begin{cases} \min\limits_{P,R,\lambda} \; \operatorname{trace}(R), \\ \text{s.t. } P > 0, \; R \geq R_0, \text{ and} \\ \begin{bmatrix} aP - A^T P A & -A^T P B \\ -B^T P A & \frac{1}{m}(1-a) R - B^T P B \end{bmatrix} \geq 0, \\ \begin{bmatrix} P & -0.5\lambda c_i \\ -0.5\lambda c_i^T & \lambda b_i - 1 \end{bmatrix} \geq 0, \qquad i = 1, ..., m, \end{cases} \tag{11}$$

then the new actuator bounds $\gamma_i := (1/[R]_{ii})$, $i = 1, ..., m$, enforce that the resulting reachable set $\mathcal{R}$ does not intersect with the dangerous states $\mathcal{D}$.

The proof of Theorem 1 can be found in Appendix A

Remark 1. In Proposition 1 and Theorem 1 there is an unknown parameter a that enters nonlinearly with the variable P that quantifies an approximation of the reachable set. To side-step the nonlinearity that is caused by the product aP, it is possible to perform a grid-search or bisection over the parameter "a," such that the optimization is repeated a number of times for different values of a. The solution, P, that corresponds to the bounding ellipsoid with minimum volume is the one that is ultimately kept.

To solve the optimization problem in Equation (11) we use YALMIP with the SDPT3 solver. An example of the code to find the optimal bounds is presented next:

Listing 1. Code to compute the optimal bounds

```
a=0.99;
P=sdpvar(n,n,'symmetric');
R=sdpvar(m,m,'diagonal');
L=sdpvar(1,1);
Co=(P>=0);
Co=Co+(R>=0);
Co=Co+([a*P-A'*P*A -A'*P*B;
        -B'*P*A (1-a)*R-B'*P*B]>=0);
for i=1:k
Co=Co+([P -0.5*L*c(i,:);
       -0.5*L*c(i,:)' L*b(i)-1]>=0)
end
sol=optimize(Co,trace(R),ops)
```

Notice that we only need a few lines of code to formulate the entire optimization problem and find the optimal solution. Also, notice that Theorem 1 is completely independent of time, given that the approximation of the reachable set considers any attack sequence, even if it is infinite.

Remark 2. Given that P is symmetric, and R is diagonal, the number of decision variables is given by $n(n+1)/2 + m$, where m is the number of actuators and n is the number of states of the system model. Since the problem is convex and linear, infeasible primal-dual path-following interior-point algorithms (the one used by SDP3) can efficiently solve the optimization problem in polynomial time $\mathcal{O}(nL)$. [37], [38]. Given that our optimization problem is solved offline, the proposed methodology can therefore easily handle large systems.

3.3. Summary of our Proposal

We now summarize the main points of our proposal. Figure 1 outlines our security analysis and then our security design to guarantee that the physical process will behave in a safe manner, even in the presence of attacks.

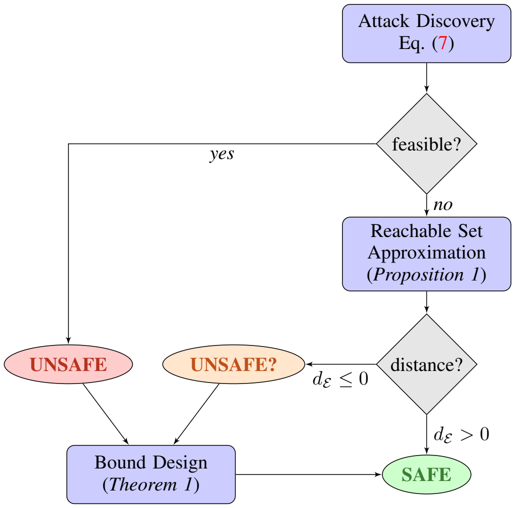

Figure 1. Flowchart of methods proposed in this paper.

Attack Discovery: Our first analysis is to find if there are feasible attacks to the system, as outlined in Equation (7). Our search algorithms are sound in that a solution is an attack which drives the system to an unsafe space; however, our attack design is not complete, in the sense that it does not guarantee that if an attack exists, our algorithms will find it.

We use this attack design problem in the next section as a baseline to compare the attacks our algorithm finds to other attacks proposed in the literature, and show that our attacks can be potentially more damaging and sometimes counterintuitive (i.e., not easy to find). If we find an attack, then it is clear that the system is unsafe in its current design, and then we have to use Theorem 1 to design bounds for the actuators.

Reachable Set Approximation (Proposition 1): If we do not find an attack, we can use Proposition 1, which uses our new method to compute the outer bound approximation of the reachable sets. If we find that the distance from our ellipsoid approximation to the unsafe states is greater than zero, then we can formally conclude that the system is safe, even under attacks.

While a positive distance guarantees that the system is safe, a negative (or zero) distance, on the other hand does not give us a guarantee that the system is unsafe. Proposition 1 gives us an outer approximation to the reachable states, but it is a conservative approximation that guarantees that the behavior of the system cannot leave the computed ellipsoid, but does not guarantee that all states in the ellipsoid can be reached. As a result we cannot prove the system is either safe or unsafe, and therefore we have to take a look at Theorem 1 to design a new system that will be secure.

Bound Design (Theorem 1): If the system is unsafe (as determined by the attack design problem), or if we cannot prove the system is safe (with Proposition 1), we can turn to Theorem 1. By designing new bounds to the actuators of the system, we can change the system to formally prove the system is safe. Notice that this bound design problem is always feasible (assuming we start in a safe state) because the possibility of having all control bounds equal to zero (forcing the system to not have any external input) will maintain the system in a safe condition at all times. Theorem 1 guarantees the largest limits that keep our ellipsoid approximation from intersecting the unsafe region.

In the next section we will show how we use these three tools (attack discovery, outer bound approximation, and the design of physical limits) to identify threats to the system, and design new systems that are secure against these powerful new attacks.

4. Case Studies

In this section we show how to apply our theoretical results to practical problems.

4.1. Case Study 1: Quadruple Tank Process

The quadruple tank process [39] is a benchmark process in control systems and more recently also in cybersecurity for control systems. The process consists of four tanks, two pumps, and two water sensors as illustrated in Figure 2. The main goal is to control the level of water in the lower two tanks by controlling the amount of water injected by the pumps. What makes this process complex is the cross interaction among the upper tanks, the lower tanks, and the pumps. In particular, if Pump 1 increases its water inlet flow (increase in v_1), it will increase the water level of Tank 1 and Tank 4; at the same time, since the level of water in tank 4 is increasing, it will affect the water flowing from Tank 4 to Tank 2. The same happens with Pump 2 and Tanks 2,3 and 1.

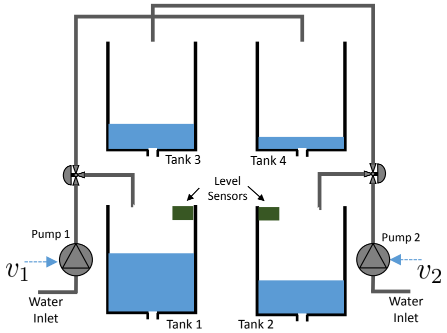

Figure 2. Quadruple Tank Scheme

The actuators are the pumps and the sensors report the water level of Tanks 1 and 2, which we call y_1, y_2. Let g represent gravity, A_i the cross section of Tank i, a_i cross-section of the outlet hole, and h_i the water level of Tank i. The control to Pump i is v_i and the corresponding flow is proportional to the control value $k_i v_i$. The parameters $\rho_1, \rho_2 \in (0, 1)$ indicate the setting of the valve that distributes the amount of water that each tank receives from the pumps. The flow to Tank 1 is $\rho_1 k_1 v_1$ such that the flow of Tank 4 is $(1-\rho_1)k_1 v_1$. Similarly for Tanks 2 and 3. We can define h_i^0 as the desired water level for each tank, and v_i^0 the necessary voltage to maintain the water level in h_i^0. Let $x_i = h_i - h_i^0$ and $u_i = v_i - v_i^0$.

The process dynamics can be modeled with the following set of equations:

$$\begin{aligned} \dot{x}(t) &= Fx(t) + Gx(t) \\ y(t) &= Hx(t), \end{aligned} \tag{12}$$

where

$$F = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix},$$

$$G = \begin{bmatrix} \frac{\rho_1 k_1}{A_1} & 0 \\ 0 & \frac{\rho_2 k_2}{A_2} \\ 0 & \frac{(1-\rho_2)k_2}{A_3} \\ \frac{(1-\rho_1)k_1}{A_4} & 0 \end{bmatrix}, \; H = \begin{bmatrix} k_c & 0 & 0 & 0 \\ 0 & k_c & 0 & 0 \end{bmatrix},$$

$T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_i^0}{g}}$ for $i = 1, \ldots, 4$.

Because we are interested in cyber-physical systems that are controlled by computers (not analog devices like control systems used to operate some years ago) we need to obtained the discrete time equation (1) with matrices A, B by using the methodology in Appendix B with a sampling period of $\tau = 0.4\ s$. The parameter values of the process are given in the following table.

A_1, A_3	$28\ cm^2$
A_2, A_4	$32\ cm^2$
a_1, a_3	$0.142\ cm^2$
a_2, a_4	$0.114\ cm^2$
k_c	$0.5\ V/cm$
g	$981\ cm/s^2$
h_1^0, h_2^0	$12\ cm$
h_3^0, h_4^0	$(1.43, 8)\ cm$
v_1^0, v_2^0	$(8.6, 3.2)\ V$
k_1, k_2	$(3.33, 3.33)\ cm^3/sV$
ρ_1, ρ_2	$(0.5, 0.3)$

The goal of the attacker is to make at least one tank overflow. Since the height of each tank is 20 cm, the set of dangerous states can be represented according to Equation (2) with four half spaces with the following parameters:

$$c_1 = [1\ 0\ 0\ 0],\ b_1 = 20 - h_1^0 = 8 \tag{13}$$
$$c_2 = [0\ 1\ 0\ 0],\ b_2 = 20 - h_2^0 = 8 \tag{14}$$
$$c_3 = [0\ 0\ 1\ 0],\ b_3 = 20 - h_3^0 = 18.57 \tag{15}$$
$$c_4 = [0\ 0\ 0\ 1],\ b_4 = 20 - h_2^0 = 12 \tag{16}$$
$$\tag{17}$$

which represent the cases where the water level reaches the capacity of the tanks.

Now that we have identified the model, the unsafe states and the desired operational points, we can solve the optimization problem in Section 3.2 to find the optimal constraints that we should impose to the pumps in order to guarantee that no attack can drive the states to unsafe states.

Experiment 1: Original Operating Range of Pumps. We assume an adversary is able to launch a man-in-the-middle attack between the PLC and the pumps as previously demonstrated in a similar water tank control system [40]. The attack starts starting at time $t_a = 150\ s$ by intercepting each fieldbus packet containing the control commands, or sensor readings and modifies the payload. We test four different types of attacks: i) Actuator bias attack, where $u_i := u_i + \delta_i$, with $\delta = [10, -10]$; ii) Actuator random attack, with $u_i \sim N(0, 10)$, i.e., u_i is drawn from a normal distribution with mean 0 and variance 10; iii) Sensor scaling attack, where $y := (t - t_a)y$, such that the scaling factor increases with time; iv) our Actuator optimal attack, where the objective is to cause an overflow in Tank 1 and Tank 2 while minimizing the attack visibility (i.e., trying to remain stealthy by keeping the attack as small as possible).

The optimization problem for the optimal attack is derived from the formulation in Equation (6)

$$\begin{aligned} &\min_{\boldsymbol{a}} \mathcal{O} = \|\boldsymbol{a}\| \\ &s.t. \\ &\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} (A^{\mathcal{T}} x_0 + H\boldsymbol{a}) = \begin{bmatrix} 8 \\ 8 \end{bmatrix} \\ &-8.6 \leq a_{1,t} \leq 3.4 \;\text{ for all } t = 0, 1, \ldots, \mathcal{T} - 1 \\ &-3.2 \leq a_{2,t} \leq 8.8 \;\text{ for all } t = 0, 1, \ldots, \mathcal{T} - 1 \end{aligned} \tag{18}$$

where the objective function $\mathcal{O} = \|\boldsymbol{a}\|$ minimizes the attack input to reach the overflow of Tanks 1 and 2, and $\mathcal{T} = 200$. Notice that in this case we are only interested in overflowing Tanks 1 and 2, so that we pre-multiply equation (5) by $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$. The operational range of each pump is $0-12$, such that the constraints in our model are $0 - v_i^0 \leq a_{i,t} \leq 12 - v_i^0$ for $i = 1, 2$.

Figure 3 illustrates the dynamic behavior of each water level h_i and the pump v_i for the different attacks. Notice that the first 3 attacks do not require any kind of knowledge about the system and only the bias attack is able to overflow Tank 1. The random attack and scaling attack do not have a significant impact in the system. On the other hand, our proposed optimal attack is able to find any feasible attack trajectory to drive both tanks to overflow in a very specific time while minimizing the change in the control action. Figure 4 shows the approximation of the reachable set with the original bounds according to Proposition 1. Notice that the outer approximation overlaps with the dangerous states such that it is not possible to guarantee that the system is secure. In fact, the bias attack and the optimal actuator cause the system to reach the unsafe region causing some tanks to overflow. While in this use-case we found a feasible attack without the problem formulated in Section 3.1, in the next section we will show that there are cases where designing an attack to reach unsafe regions is not straightforward, and we need to use the tools introduced for our *counterexample* attack.

Experiment 2: Optimal Defense. Now, applying our proposed defense strategy, we can change the original pumps by purchasing new pumps with a different operat-

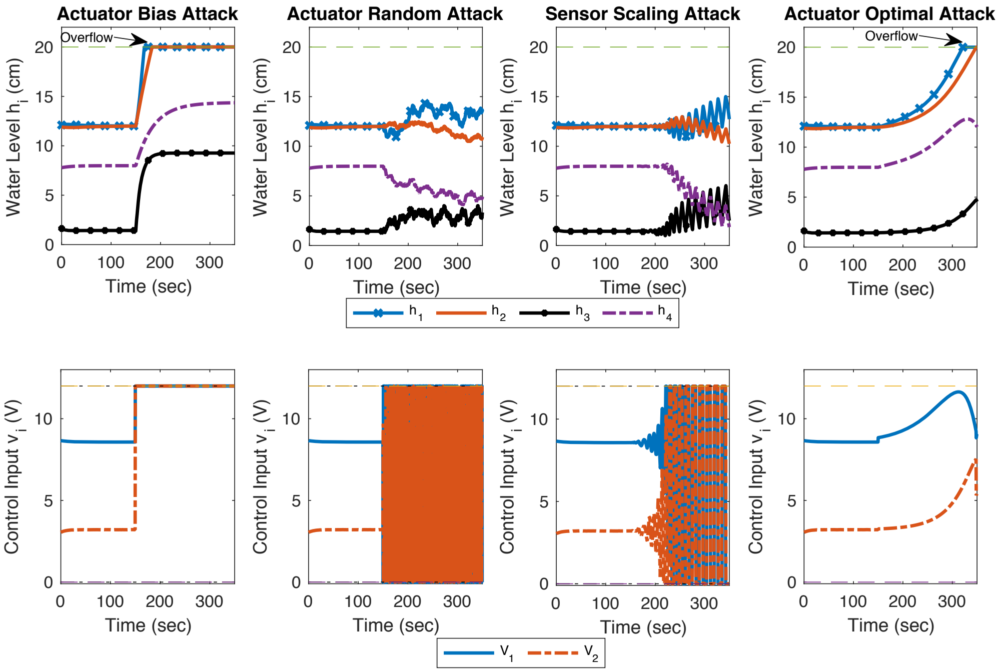

Figure 3. Water level h_i and control input v_i for the quadruple-tank process for 4 different attacks with the original actuation bounds. After 150 s, an attack is launched and it has a duration of 200 s. Notice that only the bias attack and the optimal attack are able to overflow at least one of the tanks.

ing range. To find the right operating range we need to solve the optimization problem in Equation (11). Using the code in Listing 1 we find $R = \text{diag}(0.457, 0.37)$, such that $\gamma_1 = 2.19$ and $\gamma_2 = 2.7$. As a consequence, the new safe operating range correspond to $-1.48 \leq u_1 \leq 1.48$, and $-1.65 \leq u_2 \leq 1.65$.

We now launch the same attacks as in Experiment 1, and the results are shown in Figures 5 and 6. In this case, no attack is able to overflow the tanks, and our last attack performed a counterexample search over various time lengths $\mathcal{T} = 200$, $\mathcal{T} = 500$, $\mathcal{T} = 5000$:

$$\begin{aligned}
&\min_{\boldsymbol{a}} \mathcal{O} = \|\boldsymbol{a}\| \\
&s.t. \\
&\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} (A^{\mathcal{T}} x_0 + B\boldsymbol{a}) = \begin{bmatrix} 8 \\ 8 \end{bmatrix} \\
&-1.48 \leq a_{1,t} \leq 1.48 \quad \text{for all } t = 0, 1, \ldots, \mathcal{T}-1 \\
&-1.65 \leq a_{2,t} \leq 1.65 \quad \text{for all } t = 1, 2, \ldots, \mathcal{T}-1
\end{aligned} \tag{19}$$

As expected, the solver was not able to find any feasible solution because the system was designed to be secure. Our new selected operating range for the pumps guarantee that there is no control sequence that can drive the system to dangerous states. Figure 6 shows the outer approximation of the secure reachable set. Notice that the ellipsoid does not intersect with the unsafe states, such that we can guarantee that the system is secure, and no attack can drive the water level to overflow.

4.2. Case Study 2: Vehicular Platooning

In this case study, we consider a system of four cooperating autonomous vehicles that form a vehicular platoon [41], as illustrated in Figure 7. In our system, the vehicles use on-board sensors (e.g., lidar) to maintain a given distance and the cooperate to form a platoon [42]. This cooperative signal is modeled by an additive acceleration term sent over wireless communication.

We use the model of one of the early papers studying platooning security [43]. In their model, the dynamics of the positions and velocities of the vehicles are described with the following differential equations,

$$\begin{cases}
\dot{x}_1 = v_1 \\
\dot{x}_2 = v_2 \\
\dot{x}_3 = v_3 \\
\dot{v}_1 = \quad k_p(x_2 - x_1 - d^*) + k_d(v_2 - v_1) + \beta v_1 + u_1 \\
\dot{v}_2 = -k_p(x_2 - x_1 - d^*) - k_d(v_2 - v_1) \\
\qquad +k_p(x_3 - x_2 - d^*) + k_d(v_3 - v_2) + \beta v_2 + u_2 \\
\dot{v}_3 = -k_p(x_3 - x_2 - d^*) - k_d(v_3 - v_2) \\
\qquad +k_p(x_4 - x_3 - d^*) + k_d(v_4 - v_3) + \beta v_3 + u_3 \\
\dot{v}_4 = -k_p(x_4 - x_3 - d^*) - k_d(v_4 - v_3) + \beta v_4 + u_4
\end{cases} \tag{20}$$

where $k_p = 2$ and $k_d = 1.5$ are the proportional and derivative gains of an on-board Proportional-Derivative (PD) controller, which regulates the distance between neighboring vehicles to be the desired distance $d^* = 2$ m; $\beta = -0.1$ characterizes the loss of velocity as a result of friction; and u_i with $i \in \{1, 2, 3, 4\}$ are feedforward inputs (acceleration) added to each vehicle. In cooperative cruise control settings, such feedforward inputs are used to optimize the performance of the platoon by each vehicle

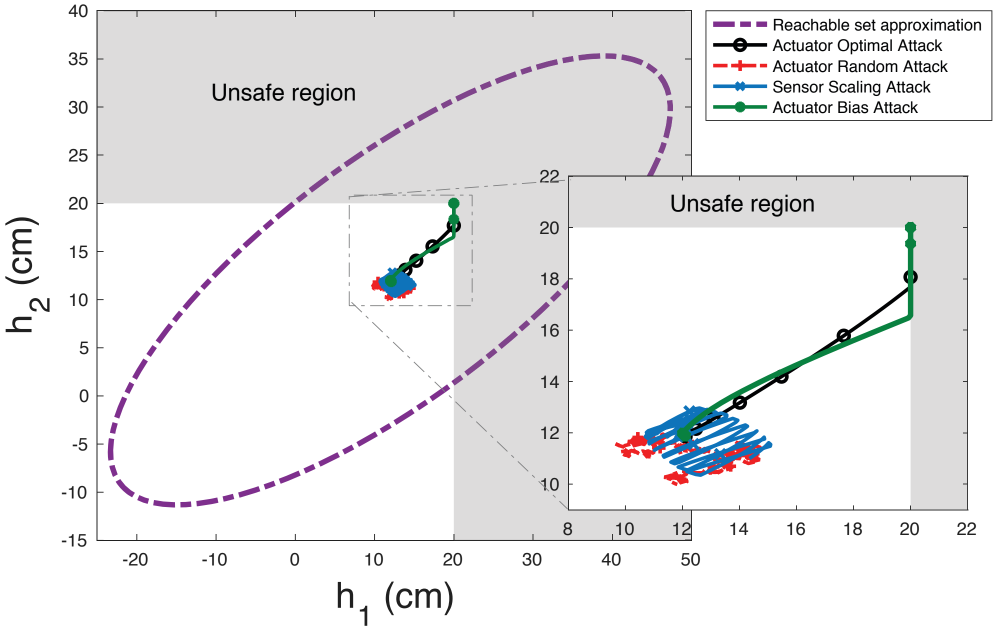

Figure 4. Trajectories of the water level h_i for the quadruple-tank process for 4 different attacks. The dashed ellipsoid represents the estimation of the reachable set with the original actuation bounds and the unsafe region represents the overflow. Notice that the bias attack and the optimal attack are able to overflow at least one of the tanks.

sharing its intended maneuvers, thus requiring the PD control to only compensate for errors. In this setting, we illustrate that if these inter-vehicle communications are compromised it is possible for attackers to crash vehicles they do not compromise. Furthermore, in this work we consider imposing bounds on the allowable feedforward inputs applied to the vehicles in order to ensure the safety of the platoon, where safety is defined as avoiding crashes between any vehicles.

The platoon is most concisely described by the *relative* distances between each pair of adjacent vehicles, defined as $d_{12} = x_2 - x_1$, $d_{23} = x_3 - x_2$, and $d_{34} = x_4 - x_3$. We can introduce new relative distance error variables $e_{12} = d_{12} - d^*$, $e_{23} = d_{23} - d^*$, and $e_{34} = d_{34} - d^*$ and rewrite the Equation (20) in terms of seven state variables $x = [e_{12}, e_{23}, e_{34}, v_1, v_2, v_3, v_4]^T$ such that

$$\dot{x}(t) = Fx(t) + Gu(t)$$

with input $u = [u_1, u_2, u_3, u_4]^T$, $G = [0_{4\times3}, I_4]^T$, and

$$F = \begin{bmatrix} 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 1 \\ k_p & 0 & 0 & \beta - k_d & k_d & 0 & 0 \\ -k_p & k_p & 0 & k_d & \beta - 2k_d & k_d & 0 \\ 0 & -k_p & k_p & 0 & k_d & \beta - 2k_d & k_d \\ 0 & 0 & -k_p & 0 & 0 & k_d & \beta - k_d \end{bmatrix},$$

where I_4 is the 4×4 identity matrix. As in the prior example, this continuous-time differential equation is sampled at discrete units of time as discussed in Appendix B using Equation (29) with time step $\tau = 0.1$s to yield a discrete-time linear time invariant dynamical system of the form in Equation (1).

Potential attackers generally intend to disturb the existing coordination between the vehicles in the platoon. In this study we assume that the communication to the leading truck (vehicle 4) and the last truck (vehicle 1) have been hijacked by the attackers, i.e., the attackers can falsify (completely determine) the inputs u_1 and u_4.

We consider that the objective of the attacker is to use vehicles 1 and 4 to cause a crash between the two other interior trucks of the platoon (vehicles 2 and 3) while avoiding a crash in the vehicles it has compromised.

Assuming the input with natural bounds $\gamma_1 = \gamma_4 = 4$ m/s^2, a trivial strategy for the attackers is to use the maximum capacity of the inputs to accelerate vehicle 1 with $u_1 = 4$ m/s^2 and decelerate vehicle 4 with $u_4 = -4$ m/s^2. This “bias” attack will collapse the platoon, including the distance between vehicles 2 and 3 because they are sandwiched between vehicles 1 and 4. This strategy and its outcome is depicted as Case 1 in Figure 8, crashing the platoon in only 2 seconds. We can use Proposition 1 to compute an outer ellipsoidal approximation of the reachable set (see Figure 9, left plot), which reinforces that there are likely to be many reachable states that correspond to a crash between vehicles of the platoon (we build a reachable set using the feasibility optimization proposed in this paper and see that indeed there are many reachable states that fall within the dangerous states). We found that the minimum volume ellipsoid corresponded to a value of $a = 0.9831$.

Hoping to secure the system, we reduce the input bounds to $\gamma_1 = \gamma_4 = 2.3$ m/s^2. In this case, the simple bias attack strategy of decelerating the leading vehicle and accelerating the last vehicle does not work with these lowered bounds, even if the attackers prolong the attack (see Case 2 in Figure 8). Although the bias attack does not cause a crash, it does not necessarily mean that the system is safe. In fact, by looking at the reachable set

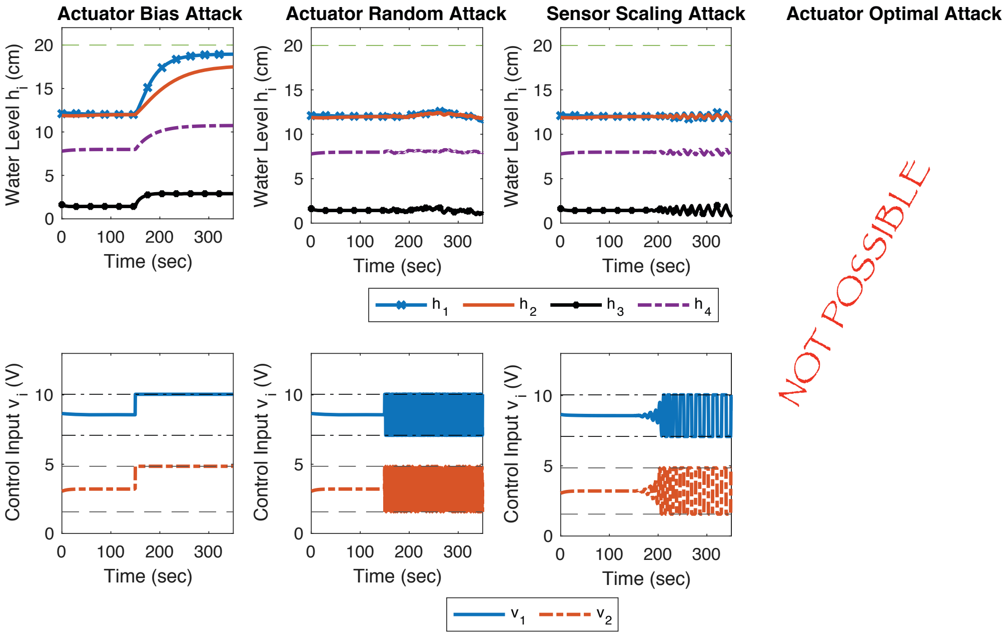

Figure 5. Water level deviation h_i and control input v_i for the quadruple-tank process for 4 different attacks with the secure bounds obtained according to Theorem 1. Notice that no attack can drive the tanks to overflow. Also, since we are guaranteeing that our system is secure, our counterexample search is unfeasible (rightmost column).

corresponding to these reduced bounds (see Figure 9, middle plot), we see that there are attack sequences that can drive the system to reach the dangerous states (i.e., cause a crash). We can now use the optimization approach to generate the attack and corresponding trajectory that causes this crash between vehicles 2 and 3 ($d_{23} = 0$ m).

Attackers aim to reach $e_{23} = -2$ m which is equivalent to a crash between vehicles 2 and 3 in t time steps. The state of the system after t time steps can be computed by equation (6). So we can define a feasibility problem:

$$\begin{aligned} &\min_{U} 0 \\ &s.t. \\ &-2 = [0,1,0,0,0,0,0]^T(A^t x_0 + HU) \qquad (21) \\ &u_j \in \mathcal{U}, \quad \text{for all } j = 1,2,\ldots,t-1 \end{aligned}$$

where $\mathcal{U}$ is the set of input signals satisfying the bounds $\gamma_1 = \gamma_2 = 2.3$ m/s^2 and x_0 is the state of the system when attack starts. In this scenario, we assumed that the platoon has reached the desired distance before the attack and all vehicles are moving with the same speed of 30 m/s. Thus, the initial state is $x_0 = [0,0,0,30,30,30,30]^T$. Since the attackers goal only focuses on the distance between vehicles 2 and 3, we only set the equality constraint in our feasibility problem such that $d_{23} = 0$ at final time step.

This feasibility problem was solved using YALMIP and the SDPT3 solver and resulted in a series of inputs $\{u_0, \ldots, u_{t-1}\}$ which can drive the system to crash in $t = 80$ time steps (see Case 3 in Figure 8).

We now select the limits to make the system secure by using Theorem 1. As discussed before, the attackers attempt to make a crash between vehicles, so we can define dangerous states as the areas in the state-space where the distances become less than or equal to zero ($d_{12}, d_{23}, d_{34} \leq 0$ m) which is equivalent to $e_{12}, e_{23}, e_{34} \leq -2$ m. Hence, we can represent the dangerous states as the union of three half spaces defined by Equation (2):

$$\begin{aligned} c_1 &= \begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, \quad b_1 = -2 \\ c_2 &= \begin{bmatrix}0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}, \quad b_2 = -2 \qquad (22) \\ c_3 &= \begin{bmatrix}0 & 0 & 1 & 0 & 0 & 0 & 0\end{bmatrix}, \quad b_3 = -2 \end{aligned}$$

In this scenario, the attackers only have access to the inputs u_1 and u_4, so the input matrix of the system will be

$$G = \begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1\end{bmatrix}^T \qquad (23)$$

We can now formulate the optimization problem described by Equation (11). As all constraints are linear inequalities and the objective function, which is the trace of diagonal matrix R, is also convex, this optimization problem is a convex programming problem [44]. We solved this problem using SDPT3 which resulted in the following bounds:

$$\gamma_1 = \gamma_2 = 1.58 \text{ m/s}^2 \qquad (24)$$

Using the same attack used in Case 3, but truncated by the new designed bounds, it can be seen that the attack is not successful (Case 4, Figure 8). In other words, with these new bounds, achieving the crash between any pair of vehicles is infeasible. For these bounds, the ellipsoidal approximation of the reachable set no longer intersects with the dangerous states (see Figure 9, right plot).

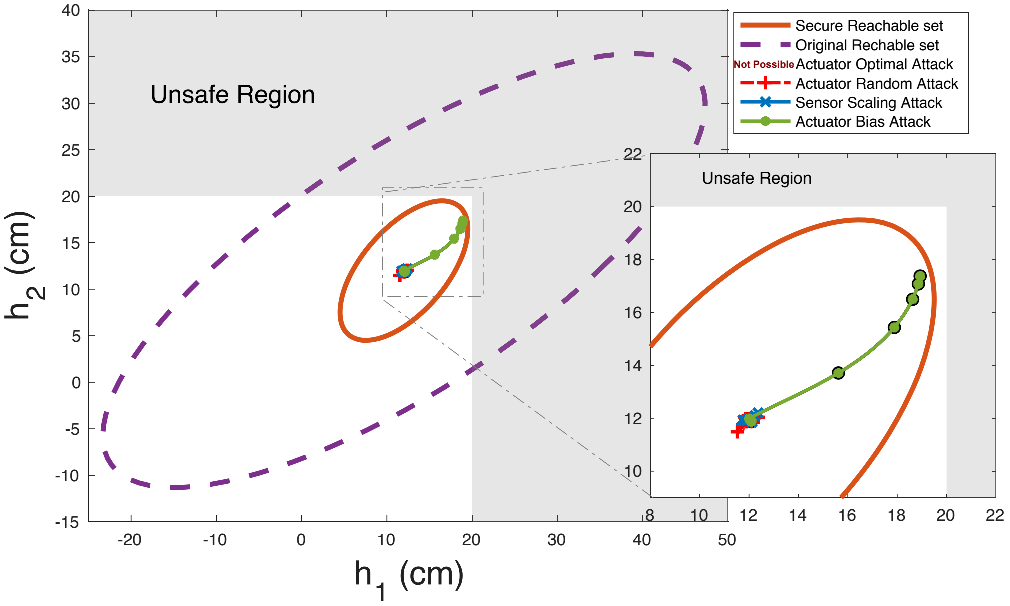

Figure 6. Trajectories of the water level h_i for the quadruple-tank process for 4 different attacks. The dashed ellipsoid represents the estimation of the reachable set with the secure bounds obtained according to Theorem 1. Notice that no attack can drive any tank to overflow. Particularly, the optimal attack becomes unfeasible since we guarantee that no attack can drive the water level to overflow.

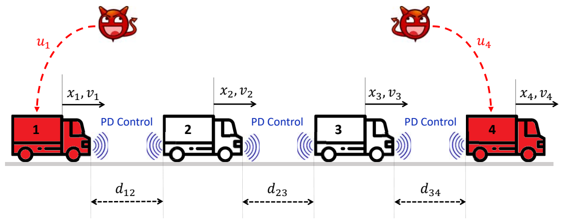

Figure 7. A platoon of 4 vehicles controlled by two separate control mechanisms. The first mechanism of control is a PD controller between each pair of neighboring vehicles, regulating the desired distance between them. The second mechanism is a cooperative control system, enabling the vehicles to communicate and can be used to assist stability of the platoon. In our scenario, attackers target the cooperative control system and gain access to inputs u_1 and u_4 to compromise vehicles 1 and 4 (shown in red).

5. Limitations and Future Work

In this paper we have presented a new security model for cyber-physical systems that allows attackers to launch completely arbitrary attacks and showed a way to design a secure system to prevent these attacks. In practice this added security will come at a cost: the control system might perform as “optimally” as desired and might result in more “sluggish” responses. While this was not an issue for our use-cases, not all cyber-physical systems might take these costs.

A way to make our approach more flexible would be to change the operating range of the actuators as a response to a detected attack. In this way our solution will only be used in emergency cases where there are attack indicators. The problem with this adaptive reconfiguration is that we may be replacing a simple hardware physical constraint with a logic one (e.g., if we implement these adaptive constraints in software in the actuator itself) which might be another target for a cyber-attack. Another possibility would be to have different sets of actuators connected to the system and activated with an analog signal sent by the attack detection algorithm.

Another adaptive solution can be done by refining our defenses dynamically in time, so if we are far away from the unsafe states, we let the actuators act normally but if we get closer to the unsafe space, we constrain the system more. We plan to explore these alternatives in future work.

On the other hand, linear approximations of nonlinear systems results in linear equations that are typically only valid in a neighborhood of an operating point, and hence, only describe the behavior of the actual system in that vicinity. Despite these limitations, linearization is still one of the most powerful tools for dealing with nonlinear systems, and it is repeatedly used successfully to model a wide variety of systems. In order to deal with multiple operating points in nonlinear systems, it is possible to define our problem as a group of linear equations, each one focused in a different operating point. As a consequence, we can extend our proposed formulation considering a system that switches among different linear subsystems (a so-called *hybrid* control system).

6. Related Work

The security of Cyber-Physical Systems and Internet-of-Things (IoT) devices has attracted significant attention in the past few years. There are various studies on the security (and insecurity) of IoT and CPS devices such as home automation, smart meters, drones, Internet-connected cameras, etc. [45]–[48].

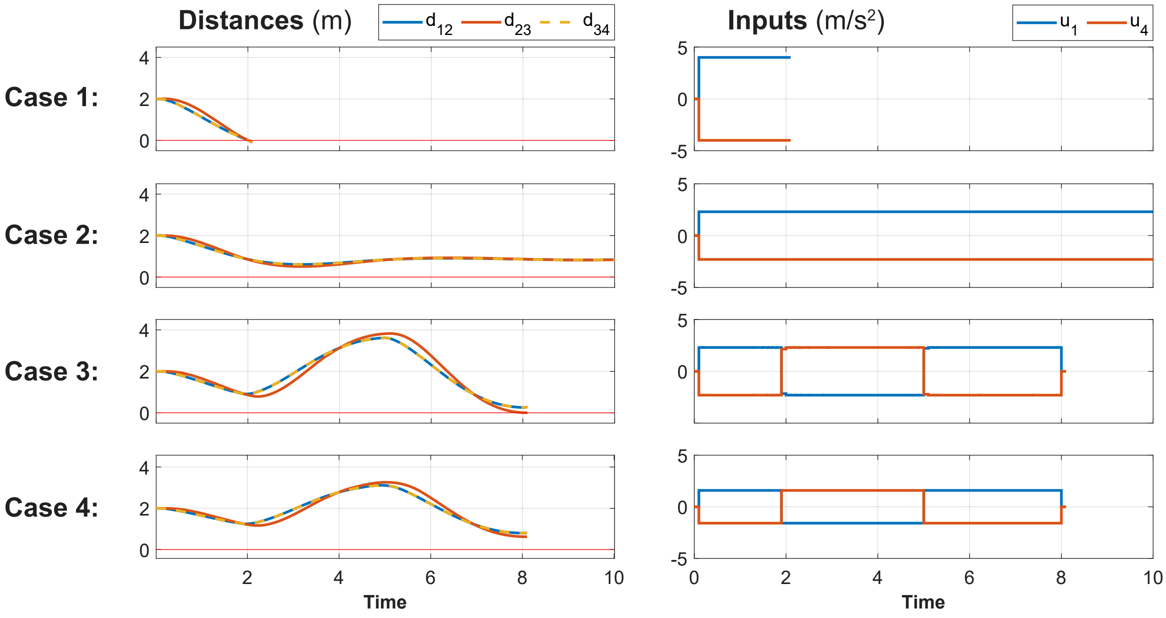

Figure 8. Attackers manipulate the feedforward signals of vehicles 1 and 4 to cause a crash between vehicles 2 and 3, i.e., by making $d_{23} \leq 0$. When the actuators work with their natural bounds ($\gamma_1 = \gamma_4 = 4$ m/s^2), attackers can perform a simple bias attack in which vehicle 4 (leading) brakes and vehicle 1 (following) accelerates, both with the maximum capacity and it can be seen that the crash happens in only 2 seconds (Case 1). By lowering the bounds from 4 m/s^2 to 2.3 m/s^2, the bias attack strategy no longer works even if allowed to run for a longer time (Case 2). However, by choosing the input signals intelligently (using the optimization approach described in this paper), the attackers are able to achieve a crash between vehicles 2 and 3 (Case 3). Using the LMI tools presented in this study, we found new actuator bounds $\gamma_1 = \gamma_4 = 1.58$ m/s^2 which guarantee the safety of the platoon. When the same intelligent strategy of Case 3 is employed, but with actuators saturated at 1.58 m/s^2, a crash does not occur (Case 4).

The ability to sense and track new physical variables has created novel security and privacy problems; for example a malicious battery can infer private information from mobile phones [49], a wearable watch can be used to infer the passwords you type [50], and new voice-enabled personal assistants can be attacked by e.g., sending voice commands stealthily embedded in songs, which, when played, can effectively control the target system through automated speech recognition without being noticed [51].

In this paper we focus on systems whose physical behavior can be changed by control commands. As mentioned in the introduction, most previous work assumes fairly limiting attacks constraining the attack time series $a(t)$ to follow predefined functions, such as scaling attacks [7], bias attacks [8], [9], maximum abrupt attacks [8]–[10], delay attacks [7], or completely random attacks [11], [12]. In addition, the security analysis of proposals show that the proposed mechanisms work to mitigate those specific attacks, but do not show how to prevent against other attacks not considered by the authors.

In contrast, this work introduces a novel secure-by-design mechanism (reducing the physical bounds of actuators) and combine it with a new proposal to approximate the reachable states of the system under (any) attack. Our methods can provably guarantee security against any attack signal.

The topic of safety and verification is also related to our work. In particular, the concept of a barrier certificate addresses the question of whether states are able to reach a set of (e.g., dangerous) states [52]. The strength of the barrier certificate method is that is generalizes easily to hybrid and nonlinear systems, however, it has primarily been used for validation/invalidation and does not have immediate ways of being adapted for the purposes of designing the system for safety or security. In addition, it requires a barrier function to be supplied, which typically constructed from a combination of experience and educated guess. If a barrier certificate can be found, we can conclude the current scenario is safe; however, if a barrier certificate cannot be found, the verification test is inconclusive.

Safety controller synthesis, on the other hand, provides a methodology to design a supervisory controller to ensure that the system avoids unsafe states [53]. However, fundamental to the operation of this safety controller is that it receives accurate observations from the system (i.e., it uses a supervisory controller that requires knowledge about the system state), and therefore it relies on the accuracy of sensors. In many cases, switching controllers are designed as safety controllers to switch between different controller modes to guarantee that the state stays in the safe region. The switching logic, however, needs to trust the sensors at every time step. In the context of attacks, where observations of the state can be falsified, this key condition may not be met—in contrast, our proposal can deal with untrusted sensors (they are a way the attacker can create the false attack signal).

In addition, the appealing aspect of our proposal–redesigning the bounds on actuation–is that it is both ag-

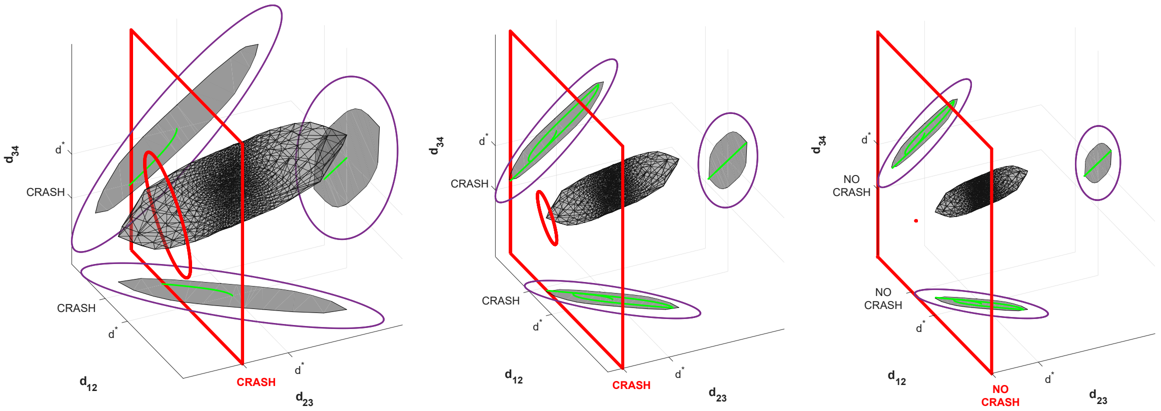

Figure 9. Projection of the 7-dimensional reachable set of the platoon into the space of state variables d_{12}, d_{23}, and d_{34} with natural bounds $\gamma_1 = \gamma_4 = 4$ m/s^2 (left), lowered bounds $\gamma_1 = \gamma_4 = 2.3$ m/s^2 (center), and synthesized safe bounds $\gamma_1 = \gamma_4 = 1.58$ m/s^2 (right). For visual clarity, we only show the 3-dimensional reachable set in three dimensions and is the wire-frame volume in each plot. We project the following objects (i.e., draw the shadow they would cast) onto the three orthogonal state planes: the reachable set (gray), the outer ellipsoidal estimations of the reachable set (purple 2D ellipses), and the system's trajectory under attack as given in Fig. 8, Cases 1, 3, and 4, respectively (green). The dangerous states ($d_{23} \leq 0$) boundary is shown by the red outlined plane; the red ellipse is the intersection of the ellipsoidal outer approximation of the reachable set with the dangerous states boundary. It can be seen that using natural bounds, a significant part of the reachable set falls in the dangerous area which means that there are inputs that attackers can use to cause a crash. Lowering the bounds from 4 m/s^2 to 2.3 m/s^2 still does not guarantee the safety as there is a small intersection between reachable set and dangerous states which enables attackers to cause a crash by choosing the input signals wisely. However, imposing the synthesized bounds of 1.58 m/s^2, the ellipsoidal approximation of the reachable set is designed to be tangent to the dangerous states. Hence, the platoon remains safe since there are not inputs that can lead to a crash.

nostic and invariant to everything that happens during the entire feedback control loop, whereas the safety controller synthesis is specifying a controller—an element in the feedback loop—to regulate the system to maintain safety.

7. Conclusions

We have introduced a new formal framework to reason about the security of cyber-physical systems. Our goal is to provide new tools to enable provable security guarantees of a control system irrespective of the attack implementation. We hope our proposal can motivate more work on attack-agnostic security solutions.

In particular we have shown how to design a CPS to prove security for arbitrary attackers, and we have also shown how to generate a feasible attack strategy. Our automatic attack feasibility search can generate new attacks that are surprising (and not easy to create). For example in the cooperative cruise control example, we found that the intuitive attack of accelerating as fast as possible did not crash the middle cars (Figure 8 case 2), but our automatically generated counterexample showed that the attacker at the back of the platoon needed to first accelerate, then break, then accelerate (Figure 8 case 3). What the attacker is creating in the platoon is a shockwave with the continuous oscillations that will destabilize the platoon and crash the cars the attacker intended. This novel attack was the result of our algorithm, and was not predefined by us as a possible strategy.

Our approach for designing the safe operating range of actuators in control systems was efficient and practical in the two use-cases we studied. Having said that, constraining the operation of a system might not work for all use-cases, in particular those that need fast response times and cannot tolerate a more sluggish response. In future work we plan to explore the trade-offs with respect to performance, security, and their associated risks.

Acknowledgments

This work was performed under the financial assistance award 70NANB17H282N from U.S. Department of Commerce, National Institute of Standards and Technology (NIST), by NSF CNS-1931573, and CNS-1929410, and by DHS/Air Force Research Laboratory under agreement number FA8750-19-2-0010. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

[1] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *Critical Infrastructure Protection*, vol. 253/2007. Springer Boston, November 2007, pp. 73–82.

[2] K. Zetter, *Countdown to Zero Day: Stuxnet and the launch of the world's first digital weapon*. Broadway books, 2014.

[3] A. Cherepanov, "Win32/industroyer, a new threat for industrial control systems," *White Paper. ESET*, 2017.

[4] R. M. Lee, M. J. Assante, and T. Conway, "German steel mill cyber attack," *Industrial Control Systems*, vol. 30, p. 62, 2014.

[5] AP, "Revenge hacker: 34 months, must repay georgia-pacific $1m," https://www.usnews.com/news/louisiana/articles/2017-02-16/revenge-hacker-34-months-must-repay-georgia-pacific-1m, February 2017.

[6] B. Johnson, D. Caban, M. Krotofil, D. Scali, N. Brubaker, and C. Glyer, "Attackers deploy new ICS Attack Framework" TRITON" and cause operational disruption to critical infrastructure," *Threat Research Blog*, 2017.

[7] R. Tan, V. Badrinath Krishna, D. K. Yau, and Z. Kalbarczyk, "Impact of integrity attacks on real-time pricing in smart grids," in *Proceedings of the 2013 ACM SIGSAC conference on Computer & communications security*. ACM, 2013, pp. 439–450.

[8] Y. Chen, C. M. Poskitt, and J. Sun, "Learning from mutants: Using code mutation to learn and monitor invariants of a cyber-physical system," in *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2018, pp. 648–660.

[9] H. Choi, W.-C. Lee, Y. Aafer, F. Fei, Z. Tu, X. Zhang, D. Xu, and X. Deng, "Detecting attacks against robotic vehicles: A control invariant approach," in *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, ser. CCS '18. New York, NY, USA: ACM, 2018, pp. 801–816.

[10] C. Feng, V. R. Palleti, A. Mathur, and D. Chana, "A systematic framework to generate invariants for anomaly detection in industrial control systems," in *2019 Network and Distributed System Security Symposium (NDSS)*.

[11] Y. Wang, Z. Xu, J. Zhang, L. Xu, H. Wang, and G. Gu, "Srid: State relation based intrusion detection for false data injection attacks in scada," in *European Symposium on Research in Computer Security*. Springer, 2014, pp. 401–418.

[12] R. G. Dutta, F. Yu, T. Zhang, Y. Hu, and Y. Jin, "Security for safety: a path toward building trusted autonomous vehicles," in *Proceedings of the International Conference on Computer-Aided Design*. ACM, 2018, p. 92.

[13] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2009, pp. 911–918.

[14] A. Hoehn and P. Zhang, "Detection of replay attacks in cyber-physical systems," in *2016 American Control Conference (ACC)*. IEEE, 2016, pp. 290–295.

[15] Z. Guo, D. Shi, K. H. Johansson, and L. Shi, "Optimal linear cyber-attack on remote state estimation," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 4–13, 2017.

[16] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2016, pp. 1092–1105.

[17] D. Hadžiosmanović, R. Sommer, E. Zambon, and P. H. Hartel, "Through the eye of the plc: semantic security monitoring for industrial processes," in *Proceedings of the 30th Annual Computer Security Applications Conference*. ACM, 2014, pp. 126–135.

[18] M. Caselli, E. Zambon, J. Amann, R. Sommer, and F. Kargl, "Specification mining for intrusion detection in networked control systems," in *25th {USENIX} Security Symposium ({USENIX} Security 16)*, 2016, pp. 791–806.

[19] S. Nürnberger and C. Rossow, "Vatican-vetted, authenticated can bus," in *Conference on Cryptographic Hardware and Embedded Systems (CHES)*, 2016.

[20] J. Van Bulck, J. T. Mühlberg, and F. Piessens, "Vulcan: Efficient component authentication and software isolation for automotive control networks," in *Proceedings of the 33rd Annual Computer Security Applications Conference*. ACM, 2017, pp. 225–237.

[21] R. G. Dutta, X. Guo, T. Zhang, K. Kwiat, C. Kamhoua, L. Njilla, and Y. Jin, "Estimation of safe sensor measurements of autonomous system under attack," in *Proceedings of the 54th Annual Design Automation Conference 2017*. ACM, 2017, p. 46.

[22] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Revealing stealthy attacks in control systems," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2012, pp. 1806–1813.

[23] R. E. Kalman, "Lectures on controllability and observability," STANFORD UNIV CA DEPT OF OPERATIONS RESEARCH, Tech. Rep., 1970.

[24] ——, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[25] D. Denning, "The limits of formal security models," *National Computer Systems Security Award Acceptance Speech*, vol. 18, 1999.

[26] M. Vanhoef and F. Piessens, "Key reinstallation attacks: Forcing nonce reuse in wpa2," in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 1313–1328.

[27] C. He, M. Sundararajan, A. Datta, A. Derek, and J. C. Mitchell, "A modular correctness proof of ieee 802.11 i and tls," in *Proceedings of the 12th ACM conference on Computer and communications security*. ACM, 2005, pp. 2–15.

[28] D. Dolev and A. Yao, "On the security of public key protocols," *IEEE Transactions on information theory*, vol. 29, no. 2, pp. 198–208, 1983.

[29] J. Katz and Y. Lindell, *Introduction to modern cryptography*. CRC press, 2014.

[30] O. Goldreich, *Foundations of cryptography: volume 2, basic applications*. Cambridge university press, 2009.

[31] H. Wolkowicz, R. Saigal, and L. Vandenberghe, *Handbook of semidefinite programming: theory, algorithms, and applications*. Springer Science & Business Media, 2012, vol. 27.

[32] L. Vandenberghe and S. Boyd, "Semidefinite programming," *SIAM review*, vol. 38, no. 1, pp. 49–95, 1996.

[33] J. Lofberg, "Yalmip : a toolbox for modeling and optimization in matlab," in *2004 IEEE International Conference on Robotics and Automation (IEEE Cat. No.04CH37508)*, Sep. 2004, pp. 284–289.

[34] R. H. Tütüncü, K.-C. Toh, and M. J. Todd, "Solving semidefinite-quadratic-linear programs using sdpt3," *Mathematical programming*, vol. 95, no. 2, pp. 189–217, 2003.

[35] R. D. Monteiro, "Primal–dual path-following algorithms for semidefinite programming," *SIAM Journal on Optimization*, vol. 7, no. 3, pp. 663–678, 1997.

[36] A. A. Kurzhanskiy and P. Varaiya, "Ellipsoidal toolbox (et)," in *Proceedings of the 45th IEEE Conference on Decision and Control*. IEEE, 2006, pp. 1498–1503.

[37] C.-K. Sim, "Interior point method on semi-definite linear complementarity problems using the nesterov–todd (nt) search direction: polynomial complexity and local convergence," *Computational Optimization and Applications*, vol. 74, no. 2, pp. 583–621, 2019.

[38] K.-C. Toh, M. J. Todd, and R. H. Tütüncü, "On the implementation and usage of sdpt3–a matlab software package for semidefinite-quadratic-linear programming, version 4.0," in *Handbook on semidefinite, conic and polynomial optimization*. Springer, 2012, pp. 715–754.

[39] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Transactions on control systems technology*, vol. 8, no. 3, pp. 456–465, 2000.

[40] D. I. Urbina, J. A. Giraldo, N. O. Tippenhauer, and A. A. Cárdenas, "Attacking fieldbus communications in ics: Applications to the swat testbed." in *SG-CRC*, 2016, pp. 75–89.

[41] Y. Zheng, S. E. Li, J. Wang, L. Y. Wang, and K. Li, "Influence of information flow topology on closed-loop stability of vehicle platoon with rigid formation," in *17th International IEEE Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 2014, pp. 2094–2100.

[42] K. C. Dey, A. Rayamajhi, M. Chowdhury, P. Bhavsar, and J. Martin, "Vehicle-to-vehicle (v2v) and vehicle-to-infrastructure (v2i) communication in a heterogeneous wireless network–performance evaluation," *Transportation Research Part C: Emerging Technologies*, vol. 68, pp. 168–184, 2016.

[43] S. Dadras, R. M. Gerdes, and R. Sharma, "Vehicular platooning in an adversarial environment," in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security*. ACM, 2015, pp. 167–178.

[44] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[45] X. Feng, Q. Li, H. Wang, and L. Sun, "Acquisitional rule-based engine for discovering internet-of-things devices," in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 327–341.

[46] N. Zhang, S. Demetriou, X. Mi, W. Diao, K. Yuan, P. Zong, F. Qian, X. Wang, K. Chen, Y. Tian *et al.*, "Understanding iot security through the data crystal ball: Where we are now and where we are going to be," *arXiv preprint arXiv:1703.09809*, 2017.

[47] K. Ly and Y. Jin, "Security challenges in cps and iot: from end-node to the system," in *2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*. IEEE, 2016, pp. 63–68.

[48] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin, “Security analysis on consumer and industrial iot devices,” in *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*. IEEE, 2016, pp. 519–524.

[49] P. Lifshits, R. Forte, Y. Hoshen, M. Halpern, M. Philipose, M. Tiwari, and M. Silberstein, “Power to peep-all: Inference attacks by malicious batteries on mobile devices,” *Proceedings on Privacy Enhancing Technologies*, vol. 2018, no. 4, pp. 141–158, 2018.

[50] X. Liu, Z. Zhou, W. Diao, Z. Li, and K. Zhang, “When good becomes evil: Keystroke inference with smartwatch,” in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2015, pp. 1273–1285.

[51] X. Yuan, Y. Chen, Y. Zhao, Y. Long, X. Liu, K. Chen, S. Zhang, H. Huang, X. Wang, and C. A. Gunter, “Commandersong: A systematic approach for practical adversarial voice recognition,” in *27th {USENIX} Security Symposium ({USENIX} Security 18)*, 2018, pp. 49–64.

[52] S. Prajna, “Barrier certificates for nonlinear model validation,” *Automatica*, vol. 42, no. 1, pp. 117–126, 2006.

[53] A. Girard, “Controller synthesis for safety and reachability via approximate bisimulation,” *Automatica*, vol. 48, no. 5, pp. 947–953, 2012.

[54] N. D. That, P. T. Nam, and Q. P. Ha, “Reachable set bounding for linear discrete-time systems with delays and bounded disturbances,” *Journal of Optimization Theory and Applications*, vol. 157, pp. 96–107, 2013.

[55] S. Boyd, L. El Ghaoui, E. Feron, and V. Balakrishnan, *Linear Matrix Inequalities in System and Control Theory*, ser. Studies in Applied Mathematics. Philadelphia, PA: SIAM, 1994, vol. 15.

Appendix A. Proofs

Before introducing the proofs, we have to define the following Lemma adapted from [54]:

Lemma 1. Let V_t be a positive definite function, $V_1 = 0$, and $[ut]_i^2 \leq \gamma_i$, $i = 1, \dots, m$. Let $R = diag(\frac{1}{\gamma_1}, \dots, \frac{1}{\gamma_m})$. If there exists a constant $a \in (0, 1)$ such that the following holds, then $V_t \leq 1$:

$$V_{t+1} - aV_t - \frac{(1-a)}{m} u_t^\top R u_t \leq 0. \tag{25}$$

A.1. Proof of Proposition 1

For some positive definite matrix $P \in \mathbb{R}^{n \times n}$, let $V_t = x_t^T P x_t$ in Lemma 1. Substituting (1) and this V_t in (25) yields

$$\nu^T \underbrace{\begin{bmatrix} aP - A^T PA & -A^T PB \\ -B^T PA & \frac{(1-a)}{m} R - B^T PB \end{bmatrix}}_{Q} \nu \geq 0 \tag{26}$$

where $\nu = \left[x_t^T,\ u_t^T\right]^T$. This inequality is satisfied if and only if Q is positive semi-definite.

To ensure that the ellipsoid bound is as tight as possible, we minimize $(\det P)^{-1/2}$ since this quantity is proportional to the volume of $x_t^T P x_t = 1$. We instead minimize $\log\det P^{-1}$ as it shares the same minimizer and because for $P > 0$ this objective is convex [55] ■.

A.2. Proof of Theorem 1

The first LMI in (11) serves to construct P such that it outer bounds the reachable set of the system. This LMI comes directly from Proposition 1.

In order to ensure that the reachable set $\mathcal{R}$ avoids the dangerous states $\mathcal{D}$, a geometrical constraint can be imposed which keeps the ellipsoid $\mathcal{E}(P)$ out of the dangerous states defined by half-spaces. This geometric constraint should guarantee that all states which satisfy $x^T Px \leq 1$ also satisfy $c_i^T x \leq b_i$, $i = 1, \dots, m$. The S-procedure provides a way to combine these simultaneous inequalities [55]: these geometrical constraints are satisfied if and only if there exists a non-negative constant λ such that $(x^T Px - 1) - \lambda(c_i^T x - b_i) \geq 0$, which can be written as:

$$\begin{bmatrix} x^T & 1 \end{bmatrix} \underbrace{\begin{bmatrix} P & -0.5\lambda c_i \\ -0.5\lambda c_i^T & \lambda b_i - 1 \end{bmatrix}}_{\mathcal{V}} \begin{bmatrix} x \\ 1 \end{bmatrix} \geq 0. \tag{27}$$

The above inequality is satisfied if $\mathcal{V} \geq 0$. ■

Appendix B. Discretization of continuous-time system

Given a continuous linear time-invariant system of the form:

$$\dot{x} = Fx + Gu, \tag{28}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the input vector, $F \in \mathbb{R}^{n \times n}$ is the state matrix, and $G \in \mathbb{R}^{n \times m}$ we can find the discrete-time state space representation of the form (1) as follows:

$$\begin{aligned} A &= e^{F\tau} \\ B &= F^{-1}(e^{F\tau} - I)G, \end{aligned} \tag{29}$$

where τ is the time step used for discretization.