Actuator Security Indices Based on Perfect Undetectability: Computation, Robustness, and Sensor Placement

Jezdimir Milošević, André Teixeira, Karl H. Johansson, and Henrik Sandberg

***Abstract*—We propose an actuator security index that can be used to localize and protect vulnerable actuators in a networked control system. Particularly, the security index of an actuator equals to the minimum number of sensors and actuators that need to be compromised, such that a perfectly undetectable attack against that actuator can be conducted. We derive a method for computing the index in small-scale systems and show that the index can potentially be increased by placing additional sensors. The difficulties that appear once the system is of a large-scale are then outlined: The index is NP-hard to compute, sensitive with respect to system variations, and based on the assumption that the attacker knows the entire system model. To overcome these difficulties, a robust security index is introduced. The robust index can characterize actuators vulnerable in any system realization, can be calculated in polynomial time, and can be related to limited model knowledge attackers. Additionally, we analyze two sensor placement problems with the objective to increase the robust indices. We show that the problems have submodular structures, so their suboptimal solutions with performance guarantees can be computed in polynomial time. Finally, we illustrate the theoretical developments through examples.**

***Index Terms*—Control systems analysis, cyber-physical systems, large-scale systems, linear systems, networks, security.**

I. Introduction

Actuators are some of the most vital components of networked control systems. Through them, we ensure that important physical processes such as power production or water distribution behave in a desired way. Actuators can also be expensive, so their placement has to be carefully chosen. To place actuators in a cost-effective manner, a number of approaches have been developed [1]–[4]. However, an issue with these approaches is that they do not take security aspects into consideration. This is dangerous, since control systems can easily become a target of malicious adversaries [5]–[7]. Therefore, it is essential to check if these effective actuator placements are at the same time secure.

Motivated by this issue, we introduce novel actuator security indices δ and δ_r. These indices can be used for localizing vulnerable actuators and developing defense strategies. The security index $\delta(u_i)$ is defined for every actuator u_i, and it equals to the minimum number of sensors and actuators that need to be compromised by an attacker to conduct a perfectly undetectable attack against u_i. Since perfectly undetectable attacks do not leave any trace in the measurements [8], [9], an actuator with a small value of δ is very vulnerable. Next, we show that δ cannot be straightforwardly used in large-scale networked control systems and we introduce the robust security index δ_r to replace δ. We, then, outline properties of δ_r and propose strategies for increasing δ_r.

A. Literature Review

It has been recognized within the control community that cyber-attacks require new techniques to be handled [10]. For instance, cyber-attacks impose fundamental limitations for state estimation [11], [12], detection [13], and consensus computation [14], [15]. The most troublesome attacks are those that can inflict considerable damage and remain unnoticed by the system operator. Examples include stealthy false-data injection [16], undetectable [13], [17], and perfectly undetectable [8], [9] attacks. To characterize the vulnerability of the system and protect it against these attacks, different approaches have been proposed [18]–[20].

Our focus is on the so-called security indices. The first security index α was introduced to characterize vulnerability of sensors in a power grid [21]. Particularly, the security index $\alpha(y_i)$ of a sensor y_i equals to the optimal value of the following optimization problem:

$$\underset{x}{\text{minimize}}\ \|y\|_0 \ \ \text{subject to}\ \ y = Cx,\ y_i \neq 0. \qquad (1)$$

Here, $y \in \mathbb{R}^m$ are the sensor measurements, $x \in \mathbb{R}^n$ are the grid states, and $C \in \mathbb{R}^{m \times n}$ is the static model of the grid.

Manuscript received February 15, 2019; revised January 4, 2020; accepted March 12, 2020. Date of publication March 17, 2020; date of current version August 28, 2020. This work was supported by the Swedish Civil Contingencies Agency through the CERCES project, the Swedish Research Council, Knut and Alice Wallenberg Foundation, and the Swedish Foundation for Strategic Research. *(Corresponding author: Jezdimir Milošević.)*

Jezdimir Milošević, Karl H. Johansson, and Henrik Sandberg are with the Division of Decision and Control Systems, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, 114 28 Stockholm, Sweden (e-mail: jezdimir@kth.se; kallej@kth.se; hsan@kth.se).

André Teixeira is with the Signals and Systems, Department of Engineering Sciences, Uppsala University, 752 36 Uppsala, Sweden (e-mail: andre.teixeira@angstrom.uu.se).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2020.2981392

The first constraint imposes that attacked sensor measurements correspond to a feasible power grid state, which ensures attack stealthiness [16]. The second constraint imposes that sensor y_i is attacked. Thus, $\alpha(y_i)$ equals to the minimum number of sensors needed to attack y_i and remain stealthy. Naturally, sensors with low values of α are the most vulnerable. Once these sensors are localized, the operator can allocate additional security measures to protect them [22].

Although α proved to be a useful tool for both vulnerability analysis and development of defense strategies, there exist two issues related to this index. First, α is difficult to compute in large-scale power grids, since the problem (1) is generally NP-hard [23]. This issue is addressed in [23]–[27]. For instance, Sou *et al.* [24] proposed an upper bound on α that can be computed in polynomial time by solving the minimum s–t cut problem. This bound is also tight in several cases of interest. Second, α is defined for *static systems* and cannot be used to characterize vulnerable components in *dynamical systems*. In contrast to the first issue that is well studied, the second has been addressed only by a few works [28], [29].

The security index in [28] considerably differs from α, since it characterizes vulnerability of the entire dynamical system. In [29], a security index similar to α was introduced to characterize vulnerability of sensors and actuators within dynamical systems. In fact, α is a special case of this index [29, Sec. III.D]. However, [29] neither addressed the problems that appear in large-scale systems nor explained how this index can be used for defense purposes. In this paper, we introduce novel actuator security indices suitable for dynamical systems, tackle the challenges that appear in large-scale control systems, and propose defense strategies based on these indices.

B. Contributions

First, we propose a novel actuator security index δ. In contrast to the dynamical index from [29] that is based on the definition of *undetectability* [13], δ is based on the definition of *perfect undetectability* [9]. To calculate δ in small-scale systems, we derive a sufficient and necessary condition that compromised components need to satisfy so that we can construct a feasible point of the security index problem (Proposition 1). To prove Proposition 1, we use an algebraic condition for existence of perfectly undetectable attacks [9]. We also show that δ can potentially be increased by placing additional sensors and that placement of additional actuators may decrease δ (Proposition 2). We, then, identify three issues that appear in large-scale systems: The index δ is NP-hard to compute (Theorem 1), sensitive with respect to system variations that are expected in large-scale systems, and based on the assumption that the attacker knows the entire system model, which can be a conservative assumption in this case.

Second, we introduce the robust security index δ_r based on a structural model of the system [30]. In contrast to δ, the robust index can be calculated efficiently by solving the minimum s–t cut problem in a graph (Proposition 3). To show this, we derive a sufficient and necessary condition that compromised components need to satisfy so that we can construct a feasible point of the robust security index problem (Theorem 2). Theorem 2 is inspired by [9], where the connection between the existence of perfectly undetectable attacks and the minimum vertex separator was introduced.

The index δ_r can also be related to both the full and limited model knowledge attackers. In the context of the full model knowledge attacker, $\delta_r(u_i)$ characterizes the minimum resources for conducting a perfectly undetectable attack against u_i in any system realization. We, then, introduce an attacker with knowledge limited to a local model and measurements. We prove that he/she can also conduct a perfectly undetectable attack against u_i in any realization by compromising $\delta_r(u_i)$ components (Proposition 5). Finally, we analyze an attacker that knows only the structure of the system. In this case, $\delta_r(u_i)$ lower bounds the number of components that this attacker needs to compromise to ensure that an attack against u_i remains perfectly undetectable (Proposition 6).

Third, since the previous results imply that actuators with a small value of δ_r are potentially very vulnerable, we propose sensor placement strategies to increase δ_r. We first show that δ_r is guaranteed to increase if sensors are placed to suitable locations in the system (Theorem 3). Based on Theorem 3, we formulate two sensor placement problems with the objective to increase δ_r and show that these problems have suitable submodular structures (Proposition 7–8). This enables us to calculate suboptimal solutions of these problems with guaranteed performance efficiently. Finally, we illustrate the theoretical results through numerical examples.

The preliminary version of the paper appeared in [31]. This article differs from [31] as follows.

1) We prove that δ is NP-hard to calculate.
2) The connection of δ_r with the full and limited model knowledge attackers is derived.
3) We prove that both δ and δ_r can be increased by placing additional sensors.
4) A new section on increasing δ_r is added.
5) More detailed proofs of the results that appeared in [31] are included.
6) We extended the section with examples.

C. Organization

The remainder of this section introduces technical preliminaries. Section II introduces the security index δ. Section III investigates properties of δ. Section IV defines the robust index δ_r. Section V outlines properties of δ_r. Section VI illustrates the theoretical findings through examples. Section VII concludes the paper. Appendix contains the proofs.

D. Technical Preliminaries

1) Notation: Consider a signal $a : \mathbb{Z}_{\geq 0} \to \mathbb{R}^{n_a}$ and let $\mathcal{I}$ be a set of indices of elements of a. Then, $a \equiv 0$ means that $a(k) = 0$ for all $k \in \mathbb{Z}_{\geq 0}$; $a \not\equiv 0$ means that $a(k) \neq 0$ for at least one $k \in \mathbb{Z}_{\geq 0}$; $a_i(k)$ is the ith element of $a(k)$; $\text{supp}(a(k)) = \{i \in \mathcal{I} : a_i(k) \neq 0\}$; and $\|a\|_0 = |\cup_{k \in \mathbb{Z}_{\geq 0}} \text{supp}(a(k))|$. The normal rank of a transfer function matrix G is $\text{nrank}\, G = \max_{z \in \mathbb{C}} \{\text{rank}\, G(z)\}$ and $G^{(I)}$ is the transfer function matrix that contains the columns of G from a set I.

2) Graph Theory: Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph with a node set $\mathcal{V}$ and a set of directed edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$. We denote by $\mathcal{N}_v^{\text{in}} = \{u \in \mathcal{V} : (u, v) \in \mathcal{E}\}$ the in-neighborhood of a node v. Nodes u and v are nonadjacent if there exists no edge between them and adjacent otherwise. A directed path from v_1 to v_l is a sequence of nodes $v_1, v_2, \ldots, v_l$, where $(v_k, v_{k+1}) \in \mathcal{E}$ for every $k \in \{1, \ldots, l-1\}$. A directed path that does not contain repeated nodes is called a simple directed path. A vertex separator (resp. an edge separator) of nodes u and v is a subset of nodes $V \subseteq \mathcal{V} \setminus \{u, v\}$ (resp. edges $E \subseteq \mathcal{E}$) whose removal eliminates all the directed paths from u to v.

3) Minimum s–t Cut Problem: Let $\mathcal{G}(\mathcal{V}, \mathcal{E})$ be a directed graph, the source s, and the sink t be the elements of $\mathcal{V}$, and assume that a weight w_{uv} is associated to each edge $(u, v) \in \mathcal{E}$. A partition of $\mathcal{V}$ into V_s and $V_t = \mathcal{V} \setminus V_s$, such that $s \in V_s$ and $t \in V_t$, is called an s–t cut. We define the cut capacity by

$$C(V_s) = \textstyle\sum_{\{(u,v)\in\mathcal{E}: u \in V_s, v \in V_t\}} w_{uv}.$$

The minimum s–t cut problem can be formulated as

$$\underset{V_s}{\text{minimize}}\ C(V_s) \ \ \text{subject to } V_s \text{ and } V_t \text{ form an } s\text{–}t \text{ cut.}$$

The minimum s–t cut problem can also be interpreted as the problem of finding a minimum cost edge separator of s and t. This separator can be recovered from a solution of the problem as $E_c = \{(u, v) \in \mathcal{E} : u \in V_s, v \in V_t\}$ and its cost is $C(V_s)$.

4) Submodular Optimization: Let $\mathcal{X}$ be a finite nonempty set and $F : 2^{\mathcal{X}} \to \mathbb{R}$ be a set function. The set function F is submodular if $F(X \cup x) - F(X) \geq F(Y \cup x) - F(Y)$ holds for all $X \subseteq Y$ and $x \in \mathcal{X} \setminus Y$. F is nondecreasing if $F(X) \leq F(Y)$ holds for all $X \subseteq Y$. The following properties of submodular functions are well known [32].

Lemma 1: The sum of submodular and nondecreasing set functions is a submodular and nondecreasing set function.

Lemma 2: If F is a submodular and nondecreasing set function and $c \in \mathbb{R}$ is a constant, then $g(X) = \min\{F(X), c\}$ is a submodular and nondecreasing set function.

Many interesting problems with submodular structure can be approximately solved in polynomial time with guarantees on performance [33]. In this work, we are interested in the following two problems:

$$\underset{X}{\text{minimize}}\ |X| \qquad \text{subject to } F(X) \geq F_{\max} \tag{2}$$

$$\underset{X}{\text{maximize}}\ F'(X) \qquad \text{subject to } |X| \leq k_{\max} \tag{3}$$

where $F(\emptyset) = F'(\emptyset) = 0$, F and F' are nondecreasing and submodular, F is integer valued, and $F_{\max}, k_{\max} \in \mathbb{Z}_{\geq 0}$. Suboptimal solutions with performance guarantees for both of the problems can be obtained in polynomial time.

Lemma 3 (see [34, Th. 1]): Let X^* be a solution of (2) and $H(d) = \sum_{i=1}^{d} 1/i$. A suboptimal solution X_g of (2) that satisfies $|X_g| \leq H(\max_{x \in \mathcal{X}} F(x))|X^*|$ can be obtained in polynomial time using the algorithm given in [34, Sec. 2].

Lemma 4 (see [35, Prop. 4.3]): Let F^* be the optimal value of (3). A suboptimal solution X_g of (3) that satisfies $F'(X_g) \geq (1 - 1/e)F^*$ can be obtained in polynomial time using the algorithm given in [35, Sec. 4].

We remark that the bounds introduced in Lemmas 3 and 4 characterize the worst case performance guarantees. The algorithms mentioned in the lemmas can perform better in practice.

II. Security Index δ

In this section, we introduce the model setup and define the actuator security index δ. The plant of a networked control system is modeled by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) + B_a a(k) \\ y(k) &= Cx(k) + D_a a(k) \end{aligned} \tag{4}$$

where $x(k) \in \mathbb{R}^{n_x}$ are the plant states at time step $k \in \mathbb{Z}_{\geq 0}$, $u(k) \in \mathbb{R}^{n_u}$ are the control inputs, $y(k) \in \mathbb{R}^{n_y + n_e}$ are the sensor measurements, and $a(k) \in \mathbb{R}^{n_u + n_y}$ are the attacks.[1] We allow the last $n_e \geq 0$ elements of y to be protected, so the attacker cannot directly manipulate them. The protection can be achieved by implementing encryption/authentication schemes, and/or improving physical protection [22]. We denote by $\mathcal{X} = \{x_1, \ldots, x_{n_x}\}$ the set of states, $\mathcal{U} = \{u_1, \ldots, u_{n_u}\}$ the set of actuators, $\mathcal{Y} = \{y_1, \ldots, y_{n_y + n_e}\}$ the set of sensors, and $\mathcal{I} = \{1, \ldots, n_u + n_y\}$ the indices of elements of a.

The first n_u elements of a correspond to attacks against the actuators, while the last n_y correspond to attacks against the unprotected sensors. Therefore, B_a and D_a are given by

$$B_a = \begin{bmatrix} B & 0_{n_x \times n_y} \end{bmatrix}, D_a = \begin{bmatrix} 0_{n_y \times n_u} & I_{n_y} \\ 0_{n_e \times n_u} & 0_{n_e \times n_y} \end{bmatrix}$$

where B is assumed to have a full column rank. This is needed to exclude degenerate cases in which the attacks trivially cancel each other or cases where an actuator does not affect the system. We also adopt the following common assumption.

Assumption 1: The attacker can change the values of control inputs and measurements that correspond to attacked actuators and sensors arbitrarily, and knows the matrices A, B, C.

It is also assumed that the attacker cannot directly manipulate the nonattacked components, so the elements of a that correspond to these components are always equal to 0.

Next, we assume that the attacker wants to conduct a perfectly undetectable attack [8], [9]. Perfectly undetectable attacks are potentially very dangerous, since they do not leave any trace in the sensor measurements.

Definition 1: Let $y(k, x(0), u, a)$ indicate that the measurements at a time step k depend on an initial state $x(0)$, input u, and attack a. An attack $a \not\equiv 0$ is *perfectly undetectable* if $y(k, x(0), u, a) = y(k, x(0), u, 0)$ holds for every $k \in \mathbb{Z}_{\geq 0}$.

Due to the superposition principle that holds for linear systems, we can rewrite the measurements as

$$y(k, x(0), u, a) = y(k, x(0), u, 0) + y(k, 0, 0, a).$$

We observe that an attack $a \not\equiv 0$ is perfectly undetectable if and only if $y(k, 0, 0, a) \equiv 0$ holds. This shows that perfectly undetectable attacks can be generally analyzed without knowledge

[1] Although we focus on discrete time systems, the analysis presented in the paper can also be extended to continuous time systems.

of $x(0)$ and u. Thus, to simplify the analysis that follows, we assume that the system is in a steady state $x(0) = 0$ and $u \equiv 0$. This assumption is without loss of generality for most results in the paper, while the exceptions are clearly outlined.

We are now ready to introduce the security index δ. The security index $\delta(u_i)$ is defined for every actuator $u_i \in \mathcal{U}$ and it equals to the minimum number of sensors and actuators that need to be compromised by the attacker to conduct a perfectly undetectable attack. Additionally, u_i has to be actively used in the attack, which models a goal or intent by the attacker. Hence, the security index $\delta(u_i)$ is equal to the optimal value of the following optimization problem.

Problem 1: Calculating $\delta(u_i)$

$$\begin{aligned} \underset{a}{\text{minimize}} \quad & \|a\|_0 \\ \text{subject to} \quad & x(k+1) = Ax(k) + B_a a(k) \\ & y(k) = Cx(k) + D_a a(k) \\ & y \equiv 0, x(0) = 0 \\ & a_i \not\equiv 0. \end{aligned}$$

The objective function reflects our desire to find the minimum number of sensors and actuators to conduct a perfectly undetectable attack (sparsest signal $a : \mathbb{Z}_{\geq 0} \to \mathbb{R}^{n_u + n_y}$). The first two constraints ensure that the attack signal satisfies the physical dynamics of the system, the third constraint imposes the attack to be perfectly undetectable, and the last constraint ensures that the actuator u_i is actively used in the attack.

Before we start analyzing δ, we outline several properties of Problem 1. First, actuators with small values of δ are more vulnerable than those with large values. The worst case occurs when $\delta(u_i) = 1$. This implies that the attacker can attack u_i and stay perfectly undetectable without compromising other components. Second, Problem 1 is not always feasible. Absence of a solution implies that the attacker cannot attack u_i and remain perfectly undetectable. We, then, adopt $\delta(u_i) = +\infty$. Third, if we remove the constraint on $x(0)$ and include $x(0)$ to be an optimization variable, we recover the security index problem based on undetectable attacks [29]. Finally, the problem can be extended to capture the case where sensors and actuators are not equally hard to attack. This can be done by introducing the objective function $\sum_{j \in \mathcal{I}, a_j \not\equiv 0} c_j$, where $c_j \in \mathbb{R}^+$ would model a cost of attacking a component j.

III. Properties of δ

In this section, we show how to compute δ, that δ can be increased by placing additional sensors, and outline difficulties that appear in large-scale networked control systems. Proofs of the results from this section can be found in Appendix A.

A. Calculating δ

We first derive a sufficient and necessary condition that a set of attacked components needs to satisfy, such that we can construct an attack signal a feasible for Problem 1.

Proposition 1: Let G be the transfer function from a to y, U_a be attacked actuators, Y_a be attacked sensors, and $I_a \subseteq \mathcal{I}$ be the indices of a that correspond to U_a and Y_a. A perfectly undetectable attack conducted with U_a and Y_a in which an actuator $u_i \in U_a$ is actively used exists if and only if

$$\text{nrank}\, G^{(I_a)} = \text{nrank}\, G^{(I_a \setminus i)}. \tag{5}$$

We now discuss Proposition 1. First, we can use the condition (5) to calculate $\delta(u_i)$ as follows. We form all the subsets of attacked sensors Y_a and actuators U_a for which $u_i \in U_a$ and $|U_a| + |Y_a| = p$ hold. The initial value of p is set to 1. For each subset, we check if (5) holds, which can be done efficiently (e.g., by using the MATLAB function `tzero`). If there exists a subset for which (5) holds, then we return $\delta(u_i) = p$. Otherwise, we increase p by 1 and repeat the process.

Second, we showed in the proof that the attacker can cover an arbitrarily large attack signal injected in u_i once (5) holds. Such an attack can damage the actuator, as shown in the Stuxnet attack [6] or the Aurora experiment [36]. Additionally, since B has a full column rank, the attack necessarily results in some of the physical states x being arbitrary large. Moreover, the attack is decoupled from $x(0)$ and u, since it is constructed offline using only the model knowledge. Thus, the attack remains perfectly undetectable for any $x(0)$ and u and the assumption $x(0) = 0$ and $u \equiv 0$ is without loss of generality.

Finally, Proposition 1 helps us to avoid checking the infinite number of constraints of Problem 1. Instead, it suffices to check if the condition (5) holds for a given combination of attacked sensors and actuators.

B. Increasing δ

We now investigate how the placement of new sensors and actuators affects δ.

Proposition 2: Assume that a new component j (sensor or actuator) is placed. Let $\delta(u_i)$ (resp. $\delta'(u_i)$) be the security index of an actuator u_i before (resp. after) the placement. Then, 1) $\delta(u_i) \leq \delta'(u_i) \leq \delta(u_i) + 1$ if j is an unprotected sensor; 2) $\delta(u_i) \leq \delta'(u_i)$ if j is a protected sensor; and 3) $\delta(u_i) \geq \delta'(u_i)$ if j is an actuator.

Proposition 2 has two interesting consequences. First, it implies that we can increase δ by placing additional sensors to monitor the system. Furthermore, δ can be used to determine which sensor placement is the most beneficial. For example, one optimality criterion can be to select the placement such that the minimum value of δ is as large as possible. If the system is small scale and a small number of sensors are being placed, we can simply go through the all sensor placements and pick an optimal one. Second, Proposition 2 illustrates an interesting tradeoff between security and safety. On the one hand, to make the system easier to control and more resilient to actuator faults, more actuators should be placed in the system. On the other hand, this may decrease the security indices, so the actuators become easier to attack.

We also remark that the bounds 2) and 3) are generally not tight. Additionally, if we simultaneously place new sensors and actuators in the system, the indices can increase, decrease, or remain the same. The following example illustrates these claims.

Example 1: Let the realization of the system be

$$A = \begin{bmatrix} 0.1 & 0 \\ 0.01 & 0.1 \end{bmatrix}, B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, C = \begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix} \quad (6)$$

and assume that the sensors are not protected. Then, $\delta(u_1) = 3$ because the attacker has to compromise the sensors in addition to u_1 to remain perfectly undetectable. If we place an actuator u_2 to directly control x_2, then $\delta'(u_1) = 2$ (attacks against u_1 can be covered by manipulating u_2). If we place a protected sensor to measure x_1, then $\delta'(u_1) = +\infty$ (attacks against u_1 are always visible in the protected sensor). If we simultaneously place actuator u_2 to directly control x_2 and 1) a protected sensor to measure x_2, then $\delta'(u_1) = 2$ (same reason as above); 2) a protected sensor to measure x_1, then $\delta'(u_1) = +\infty$ (same reason as above); and 3) an unprotected sensor to measure x_1, then $\delta'(u_1) = 3$ (the attacker needs to compromise u_1, u_2, and the new sensor).

C. Large-Scale Networked Control Systems and δ

We now outline difficulties that appear once a networked control system is large scale.

1) NP Hardness of Problem 1: We showed earlier that δ can be calculated using the brute force search. However, this method is computationally intense and, therefore, inapplicable for large-scale networked control systems. In fact, Theorem 1 that we introduce next establishes that Problem 1 is NP-hard. Thus, there are no known polynomial time algorithms that can be used to solve this problem.

Theorem 1: Problem 1 is NP-hard.

Remark 1: In the proof of Theorem 1, we showed that Problem 1 can sometimes be reduced to a problem with a finite number of constraints. Nevertheless, such a problem is still NP-hard to solve due to the ℓ_0-norm in the objective.

2) Fragility of δ: Large-scale networked control systems are complex systems that can change configuration over time. For example, in a power grid, microgrids can detach from the grid [37], some power lines may be turned-off [38], or some measurements may become unavailable due to unreliable communication [39]. Unfortunately, δ can be quite sensitive with respect to changes in realization of A, B, C.

Example 2: Let the realization of the system be the same as in (6), but assume that the sensors measuring x_2 are protected. Then, $\delta(u_1) = +\infty$ because any input influences the protected outputs. However, if $A(2, 1) = 0$, the transfer function from the actuator to the sensors is 0, so $\delta(u_1) = 1$.

Lack of robustness of δ has two consequences. First, an actuator that appears to be secure in one realization of the system may be vulnerable in another. Thus, to find actuators that are vulnerable, one should calculate δ for different realizations of A, B, C. Due to NP-hardness, this is infeasible in large-scale systems. Second, even if we calculate indices for all the realizations, ensuring that δ of every actuator is large enough in every realization may require a significant budget. Naturally, we may first focus on defending those actuators that are vulnerable in any system realization. However, the question to answer is if we can find these actuators efficiently.

Remark 2: We assume that system variations occur infrequently compared to the time scale of the perfectly undetectable attacks. Hence, to the attacker, the system is linear and time-invariant.

3) Full Model Knowledge Attacker: If the system is large scale, then Assumption 1 that imposes that the attacker has the exact knowledge of A, B, C may be conservative. As illustrated in Section VI-C, lack of the full model knowledge represents a serious disadvantage for the attacker and can lead to his/her detection [40]. Thus, it is relevant to develop indices that can also be related to attackers limited to local model knowledge.

4) Replacement of δ: Due to the aforementioned three deficiencies, δ is not practical to be used in large-scale networked control systems. Therefore, we introduce the robust security index δ_r that can characterize actuators vulnerable in any system realization, can be calculated efficiently, and can be related to attackers with limited model knowledge.

IV. Robust Security Index δ_r

The robust index we introduce in this section is based on a structural model $[A], [B], [C]$ of the system [30]. The structural matrix $[A] \in \mathbb{R}^{n_x \times n_x}$ has binary elements. If $[A](i, j) = 0$, then $A(i, j) = 0$ for every realization of matrix A. If $[A](i, j) = 1$, then $A(i, j)$ can take any value from $\mathbb{R}$. Same holds for the matrices $[B] \in \mathbb{R}^{n_x \times n_u}$ and $[C] \in \mathbb{R}^{(n_y+n_e) \times n_x}$.

In the remainder, we focus on a specific case of the matrices $[B]$ and $[C]$. Particularly, we assume that each actuator directly influences only one state and each sensor directly measures only one state. These assumptions are commonly adopted in sensor and actuator placement problems for large-scale networked control systems [2], [3], [41]. Additionally, to ensure that every B has a full column rank, we assume that $[B]$ has a full column rank and exclude realizations of $[B]$ where an actuator is idle (it does not influence any state).

Assumption 2: Let e_i be the ith vector of the canonical basis of appropriate size. We assume that 1) $[B] = [e_{i_1} \dots e_{i_{n_u}}]$ and rank $[B] = n_u$; 2) if $[B](i, j) = 1$, then $B(i, j) \neq 0$ for every realization B; and 3) $[C] = [e_{j_1} \dots e_{j_{n_y+n_e}}]^T$.

Properties 1) and 2) are necessary for the derivation of the results that follow. Property 3) is introduced to simplify the presentation. The results can be generalized to the case when this property does not hold.

We now introduce an extended graph $\mathcal{G}_t = (\mathcal{V}, \mathcal{E})$ based on $[A], [B], [C]$. The node set is $\mathcal{V} = \mathcal{X} \cup \mathcal{U} \cup \mathcal{Y} \cup t$, where node t can be seen as an operator or a control center that receives the measurements from the process. The edge set is $\mathcal{E} = \mathcal{E}_{ux} \cup \mathcal{E}_{xx} \cup \mathcal{E}_{xy} \cup \mathcal{E}_{yt}$, where $\mathcal{E}_{ux} = \{(u_j, x_i) : [B](i, j) = 1\}$ are the edges from the actuators to the states, $\mathcal{E}_{xx} = \{(x_j, x_i) : [A](i, j) = 1\}$ are the edges between the states, $\mathcal{E}_{xy} = \{(x_j, y_i) : [C](i, j) = 1\}$ are the edges from the states to the sensors, and $\mathcal{E}_{yt} = \{(y_i, t) : \forall y_i \in \mathcal{Y}\}$ are the edges from the sensors to t. Since the extended graph $\mathcal{G}_t$ is crucial for

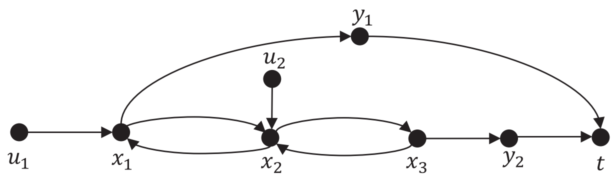

Fig. 1. Extended graph $\mathcal{G}_t$ (Example 3).

analyzing the robust index δ_r that we introduce next, we clarify it using an example.

Example 3: Let the structural matrices be given by

$$[A] = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad [B] = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad [C] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}.$$

The extended graph $\mathcal{G}_t$ is shown in Fig. 1.

Let $[A], [B], [C]$ be given and let us define a set $\mathcal{R}$ of all the system realizations (A, B, C) that are according to the model $[A], [B], [C]$ and Assumption 2. We define the robust index $\delta_r(u_i)$ of an actuator u_i as the optimal value of the following optimization problem.

Problem 2: Calculating $\delta_r(u_i)$

$$\begin{aligned} \underset{I_a \subseteq \mathcal{I}}{\text{minimize}} \quad & |I_a| \\ \text{subject to} \quad & \forall (A, B, C) \in \mathcal{R}, \exists a: \\ & \text{supp}(a) \subseteq I_a \\ & x(k+1) = Ax(k) + B_a a(k) \\ & y(k) = Cx(k) + D_a a(k) \\ & y \equiv 0, x(0) = 0 \\ & a_i \not\equiv 0. \end{aligned}$$

In words, the structural index $\delta_r(u_i)$ characterizes the minimum number of sensors and actuators that enable the attacker to attack u_i and remain perfectly undetectable in any system realization from $\mathcal{R}$. Thus, small $\delta_r(u_i)$ indicates a serious vulnerability of actuator u_i. Particularly, not just that the attacker can conduct a perfectly undetectable attack against u_i using a small number of components, but he/she can do that in any realization from $\mathcal{R}$. We also remark that Problem 2 does not have to be solvable. In that case, the attacker cannot gather components that allow him/her to attack u_i in any system realization, in which case we adopt $\delta_r(u_i) = +\infty$.

Besides the ability to characterize actuators vulnerable in any system realization, the robust index δ_r has other favorable properties that we outline next.

V. Properties of δ_r

In this section, we show that δ_r can be efficiently calculated by solving the minimum s–t cut problem, relate δ_r with the full and limited model knowledge attackers, and show how δ_r can be improved through sensor placement. Proofs of the results from this section can be found in Appendix B.

A. Calculating δ_r

We first introduce Theorem 2, which gives a sufficient and necessary condition that a set of attacked components needs to satisfy to be a feasible point of Problem 2.

Theorem 2: Let U_a be attacked actuators, Y_a be attacked sensors, u_i be an actuator from U_a, and X_a be defined by

$$X_a = \{x_j \in \mathcal{X} : (u_k, x_j) \in \mathcal{E}_{ux}, u_k \in U_a \setminus u_i\}. \quad (7)$$

A perfectly undetectable attack conducted with the components U_a and Y_a in which actuator u_i is actively used exists in any realization from $\mathcal{R}$ if and only if $X_a \cup Y_a$ is a vertex separator of u_i and t in $\mathcal{G}_t$.

The intuition behind Theorem 2 is the following. An attack against u_i can be thought of as the attacker injecting a flow into the system through u_i. To stay perfectly undetectable, he/she wants to prevent the flow from reaching the operator modeled by t. The attacker uses a strategy where he/she injects negative flows into the states X_a using the actuators $U_a \setminus u_i$, and cancels out the flows going through these states. The same applies to Y_a. If $X_a \cup Y_a$ is a vertex separator of u_i and t, then the flow is successfully canceled out, and the attack remains perfectly undetectable. However, if there exists a directed path connecting u_i and t, then we can find a realization from $\mathcal{R}$ for which the flow injected in u_i always reaches the operator.

From Theorem 2, it follows that calculating $\delta_r(u_i)$ reduces to calculating a minimum vertex separator of u_i and t consisting of X_a and Y_a. Hence, Problem 2 can be reduced to the following optimization problem:

$$\begin{aligned} \underset{U_a, Y_a}{\text{minimize}} \ & |U_a| + |Y_a| \\ \text{subject to} \ & X_a \text{ is given by } (7) \\ & Y_a \text{ contains only unprotected sensors} \\ & X_a \cup Y_a \text{ is a vertex separator of } u_i \text{ and } t \\ & u_i \in U_a. \end{aligned} \quad (8)$$

The objective reflects our goal to find a minimum size vertex separator. The first two constraints ensure that the separator consists of states X_a and unprotected sensors Y_a, the third constraint ensures that $X_a \cup Y_a$ is a vertex separator of u_i and t, and the fourth constraint imposes that u_i is compromised.

In contrast to Problem 1 that is NP-hard, the problem (8) can be reduced to the minimum s–t cut problem and solved in polynomial time using well-established algorithms [42]. To prove this claim, we first transform $\mathcal{G}_t$ to a convenient graph $\mathcal{G}_i = (\mathcal{V}_i, \mathcal{E}_i)$ with an additional set of edge weights $\mathcal{W}_i$.

Remark 3: In [9], it was explained how to construct a graph for finding a minimum vertex separator. However, in our case, not all the states can be removed and some sensors can be protected. Thus, the graph needs to be adjusted accordingly.

Let state x_j be of Type 1 if it is adjacent to an actuator from $\mathcal{U} \setminus u_i$ and Type 2 otherwise. The set $\mathcal{V}_i$ contains u_i and t (the source and the sink), $x_{j_{\text{in}}}$ and $x_{j_{\text{out}}}$ for every x_j of Type 1, and every x_j of Type 2. The sets $\mathcal{E}_i$ and $\mathcal{W}_i$ are constructed according to the following rules.

1) If $(u_i, x_j) \in \mathcal{E}_{ux}$, then $(u_i, x_j) \in \mathcal{E}_i$ and $w_{u_i x_j} = +\infty$.

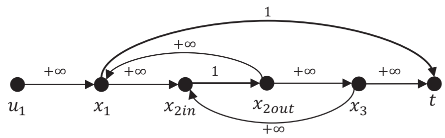

Fig. 2. Graph $\mathcal{G}_1$ (Example 4).

2) For every $(x_j, x_k) \in \mathcal{E}_{xx}, x_j \neq x_k$, we add an edge of the weight $+\infty$ to $\mathcal{E}_i$ subject to the following rules:
   a) if x_j and x_k are Type 1, then $(x_{j_{\text{out}}}, x_{k_{\text{in}}}) \in \mathcal{E}_i$;
   b) if x_j is Type 1 and x_k is Type 2, then $(x_{j_{out}}, x_k) \in \mathcal{E}_i$;
   c) if x_j is Type 2 and x_k is Type 1, then $(x_j, x_{k_{in}}) \in \mathcal{E}_i$;
   d) if x_j and x_k are Type 2, then $(x_j, x_k) \in \mathcal{E}_i$.
3) For every $x_{j_{\text{in}}}$ and $x_{j_{\text{out}}}$ that correspond to the state x_j of Type 1, $(x_{j_{\text{in}}}, x_{j_{\text{out}}}) \in \mathcal{E}_i$ and $w_{x_{j_{\text{in}}} x_{j_{\text{out}}}} = 1$.
4) For every x_j of Type 1 (resp. Type 2) that is measured, we add $(x_{j_{\text{out}}}, t)$ (resp. (x_j, t)) to $\mathcal{E}_i$. If any of the sensors measuring x_j is protected, we set the edge weight to $+\infty$. Otherwise, the edge weight equals to the number of unprotected sensors measuring x_j.

Example 4: Assume the same structural matrices as in Example 3. Let the first sensor be unprotected and the second one protected. The graph $\mathcal{G}_1$ constructed for the purpose of solving the problem (8) for actuator u_1 is shown in Fig. 2.

We now show that the optimal value of (8) can be obtained by solving the minimum u_i–t cut problem in $\mathcal{G}_i$.

Proposition 3: Let $\delta_r(u_i)$ be the robust security index of an actuator u_i and δ^* be the optimal value of the minimum u_i–t cut problem in $\mathcal{G}_i$. If $\delta_r(u_i) \neq +\infty$, then $\delta_r(u_i) = \delta^* + 1$. Otherwise, $\delta_r(u_i) = \delta^* = +\infty$ holds.

Remark 4: Proposition 3 extends the previous findings on the static security index α [24], where α was computed by solving the minimum s–t cut problem.

B. Relation of δ_r to Different Types of Attackers

We now explain how δ_r is related to the full model knowledge attacker and two limited model knowledge attackers. To distinguish between the different attackers, in the remainder, we refer to the full model knowledge attacker as Attacker 1, and to the newly introduced attackers as Attackers 2 and 3.

1) Attacker 1: As mentioned earlier, $\delta_r(u_i)$ characterizes the minimum number of sensors and actuators that enable Attacker 1 to attack u_i and remain perfectly undetectable in any realization from $\mathcal{R}$. Hence, large (resp. small) $\delta_r(u_i)$ prevents (resp. enables) Attacker 1 to easily gather disruption resources to attack u_i in any system realization. Another point worth mentioning is that $\delta_r(u_i)$ upper bounds $\delta(u_i)$.

Proposition 4: For any realization from $\mathcal{R}$ and any actuator u_i, $\delta_r(u_i) \geq \delta(u_i)$ holds. Additionally, if $\delta_r(u_i) = +\infty$, then there exists a realization from $\mathcal{R}$ in which $\delta(u_i) = +\infty$.

Unfortunately, we show in Section VI that $\delta_r(u_i)$ is not a tight upper bound of $\delta(u_i)$. Thus, there generally exist a realization in which less than $\delta_r(u_i)$ components suffices for Attacker 1 to conduct a perfectly undetectable attack against u_i. However, Attacker 1 needs to be sure that such a realization is present. If the realization occurs rarely, the attacker may need to wait for a long time, which increases his/her chances of being discovered. To avoid this, Attacker 1 may still want to compromise $\delta_r(u_i)$ components that allow him/her to conduct a perfectly undetectable against u_i in any realization from $\mathcal{R}$.

2) Attacker 2: We now show that a small $\delta_r(u_i)$ implies that u_i is vulnerable even if the attacker does not know the matrices A, B, C. Consider the following attacker.

Assumption 3: Attacker 2: 1) Can read and change the values of control inputs and measurements that correspond to attacked actuators U_a and sensors Y_a. 2) Knows $[A]$, $[B]$, $[C]$ and the rows $A(j,:)$, $B(j,:)$ that correspond to every state x_j that is adjacent to an actuator from U_a. 3) Knows for every k: $x_j(k)$ for any x_j that is adjacent to an actuator from U_a and $x_l(k)$ for any $x_l \in \mathcal{N}^{\text{in}}_{x_j}$; and 4) Wants to remain perfectly undetectable.

Attacker 2 does not know the entire realization A, B, C, but only the structural model and the rows of A and B that correspond to the attacked actuators U_a. Attacker 2 also knows the values of the states adjacent to U_a and their in-neighbors. The attacker can obtain these values by placing additional sensors, but can also get this information for free. Namely, control algorithms sometimes base decision on local and neighboring states to achieve better performance [43]. Hence, the neighboring nodes may continue sending the information to the compromised actuator nodes if the attacker remains undetected. We now relate Attacker 2 to δ_r.

Proposition 5: Let U_a be attacked actuators, Y_a be attacked sensors, u_i be an actuator from U_a, and X_a be defined as in (7). Attacker 2 can conduct a perfectly undetectable attack in which u_i is actively used in any realization from $\mathcal{R}$ if and only if $X_a \cup Y_a$ is a vertex separator of u_i and t in $\mathcal{G}_t$.

Recall that the minimum number of components that ensures $X_a \cup Y_a$ is a vertex separator of u_i and t is equal to $\delta_r(u_i) - 1$. Hence, Proposition 5 implies that Attacker 2 with the right combination of $\delta_r(u_i)$ components can conduct a perfectly undetectable attack against u_i in any realization of the system. Therefore, a small $\delta_r(u_i)$ implies that u_i is vulnerable even if the attacker does not possess the full model knowledge.

We also point out that the assumption that $x(0) = 0$ and $u \equiv 0$ is needed for this result to hold (this steady state can be substituted with any other constant steady state). Particularly, we use in the proof that Attacker 2 can construct a strategy similar to the one introduced to prove Theorem 2. However, to compensate for the lack of model knowledge, Attacker 2 exploits the steady-state assumption to implement the strategy in a feedback manner using local states and measurements. For example, we show in Section VI that if u starts changing during the attack, Attacker 2 can be revealed.

3) Attacker 3: While the previous two propositions show that a small $\delta_r(u_i)$ implies that u_i is vulnerable, a perhaps more interesting question to answer is if a large $\delta_r(u_i)$ implies that u_i is secured. Unfortunately, we cannot make such a claim, since Attackers 1 and 2 may conduct a perfectly undetectable attack against u_i with less than $\delta_r(u_i)$ components in some realizations.

Yet, we do argue that having a large $\delta_r(u_i)$ provides a reasonable level of security. Having a large $\delta_r(u_i)$ implies that attacking u_i can trigger a large number of sensors. To avoid being detected from these sensors, an attacker should make a synchronized attack using other components. Thus, he/she either needs to have a precise model and use other actuators to cancel the effect of the attack or compromise a large number of sensors. To illustrate this point, we introduce Attacker 3.

Assumption 4: Attacker 3: 1) Can read and change the values of control inputs and measurements that correspond to attacked actuators U_a and sensors Y_a. 2) Knows $[A], [B], [C]$. 3) Wants to remain perfectly undetectable.

Since Attacker 3 knows only $[A], [B], [C]$, he/she cannot constructively use other actuators to cover an attack against u_i. Namely, he/she does not know what signals to inject in attacked actuators. Yet, if the system is in a steady state, Attacker 3 can use Replay attack strategy [44] to conduct a perfectly undetectable attack against u_i. In this strategy, the attacker covers an attack against u_i by compromising sufficiently many sensors and replicating previously recorded steady-state values from these sensors.

Proposition 6 that we introduce next establishes that if Attacker 3 wants to ensure that an attack against u_i remains perfectly undetectable, then he/she needs to compromise at least $\delta_r(u_i) - 1$ sensors. Hence, a large $\delta_r(u_i)$ makes attacks against u_i more difficult for Attacker 3.

Proposition 6: Let u_i be an attacked actuator and Y_a be attacked sensors. If Attacker 3 can attack u_i and ensure the attack remains perfectly undetectable, then $|Y_a| \geq \delta_r(u_i) - 1$ holds. If $\delta_r(u_i) = +\infty$, then Attacker 3 cannot attack u_i and ensure perfect undetectability.

We further clarify Proposition 6 in an example.

Example 5: Let the structural matrices be given by

$$[A] = \begin{bmatrix} 0 & 0 \\ 1 & 1 \end{bmatrix}, \quad [B] = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad [C] = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

It can be verified that $\delta_r(u_1) = 2$. Assume that Attacker 3 targets u_1. From Proposition 6, Attacker 3 needs to compromise at least $\delta_r(u_i) - 1 = 1$ sensor to ensure an attack against u_1 remains perfectly undetectable. Indeed, let the realization be

$$A = \begin{bmatrix} 0 & 0 \\ \lambda_1 & \lambda_2 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad C = \begin{bmatrix} 0 & 1 \end{bmatrix}.$$

If $\lambda_1 \neq 0$, then any attack against u_1 is visible in the sensor. Since Attacker 3 knows only the structural model of the system, he/she does not know the exact value of λ_1. Thus, he/she needs to compromise the sensor to ensure an attack against u_1 remains perfectly undetectable.

4) Summary: The main conclusions are as follows: 1) If $\delta_r(u_i)$ is small, then u_i is vulnerable with respect to Attackers 1 and 2 in any realization from $\mathcal{R}$. 2) A large value of $\delta_r(u_i)$ does not imply security with respect to these attackers, but it prevents them from easily gathering resources for attacking u_i in any realization from $\mathcal{R}$. 3) A large $\delta_r(u_i)$ indicates security with respect to Attacker 3. For these reasons, it is useful to derive strategies for increasing δ_r that can be used in large-scale networked control systems. In the following, we consider this problem.

C. Increasing δ_r

Let u_i be an actuator for which we want to increase $\delta_r(u_i)$. Consider the extended graph $\mathcal{G}_t$ and let x_k be a state with the following properties: 1) there exists a directed path from u_i to x_k; and 2) none of the states from this path is adjacent to an actuator from $\mathcal{U} \setminus u_i$. Let the set of all such states be denoted with X_i. We show that by placing a new sensor to measure a state from X_i, the robust index $\delta_r(u_i)$ is guaranteed to increase. Moreover, if every state adjacent to an actuator is also adjacent to a sensor, then placing a new sensor to measure a state from X_i is the only way to increase $\delta_r(u_i)$.

Theorem 3: Let u_i be an actuator with $\delta_r(u_i) \neq +\infty$, X_i be defined as above, and assume that a sensor is placed to measure a state from X_i. If $\delta'_r(u_i)$ is the robust index after the placement, then $\delta'_r(u_i) = \delta_r(u_i) + 1$ (resp. $\delta'_r(u_i) = +\infty$) holds when the new sensor is unprotected (resp. protected). Additionally, if every state directly controlled by an actuator is directly measured by a sensor, then $\delta_r(u_i)$ is increased if and only if a sensor is placed to measure a state from X_i.

The sets $X_1, \ldots, X_{n_u}$ have two important properties. First, these sets are not affected by the placement of new sensors. Thus, if we place n unprotected sensors to measure states from X_i, then $\delta_r(u_i)$ is guaranteed to increase by n. Second, if we remove from $\mathcal{G}_t$ all the states that are adjacent to an actuator from $U \setminus u_i$, then X_i contains all the states to which u_i is connected with a directed path. Hence, the sets can be found using the breadth first search algorithm [45].

Next, we use the sets $X_1, \ldots, X_{n_u}$ to formulate two sensor placement problems. As we shall see, suboptimal solutions with performance guarantees of the problems can be obtained efficiently, even in large-scale networked control systems.

Remark 5: Note that increasing δ_r does not generally imply that we increase δ. However, the placement of new sensors cannot decrease δ (Proposition 2), so we definitely do not degrade this index. In fact, we illustrate in Section VI that by increasing δ_r, we may indirectly increase δ.

1) Placement of Unprotected Sensors: We first discuss the problem of placing unprotected sensors. The goal is to place these sensors to increase δ_r for every actuator u_i by some $k_i \in \mathbb{Z}_{\geq 0}$. We assume that unprotected sensors are inexpensive, so we do not have a sharp constraint on the number of sensors we should place. Yet, we still want to place the minimum number of them to achieve the desired benefit.

Let the set of sensors be $\mathcal{Y}_s = \{y_1, \ldots, y_{n_s}\}$ and x_{y_i} be the state measured by $y_i \in \mathcal{Y}_s$. For every actuator u_i, we define

$$g_i(Y_p) = \min\left\{\textstyle\sum_{y_j \in Y_p} |x_{y_j} \cap X_i|, k_i\right\}$$

where $Y_p \subseteq \mathcal{Y}_s$ is a set of newly placed sensors. This function equals k_i if at least k_i sensors from Y_p measure states from X_i. We, then, have from Theorem 3 that $\delta_r(u_i)$ increases by at least or exactly k_i. The problem we want to solve is then

$$\underset{Y_p}{\text{minimize}}\ |Y_p| \quad \text{subject to } \textstyle\sum_{u_i \in \mathcal{U}} g_i(Y_p) \geq \sum_{u_i \in \mathcal{U}} k_i. \qquad (9)$$

The objective function we are minimizing is the number of placed sensors. Additionally, if the constraint is satisfied, then

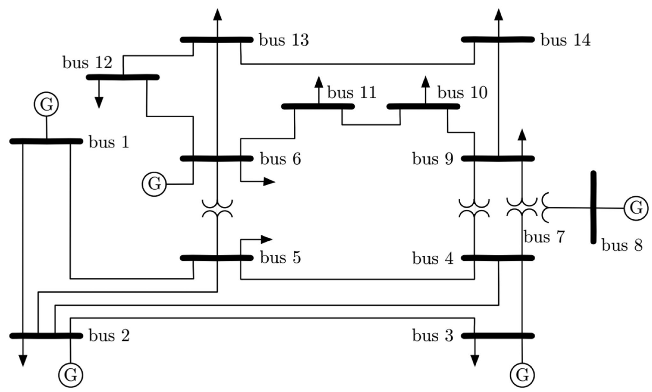

Fig. 3. Schematic of the IEEE 14-bus system [13].

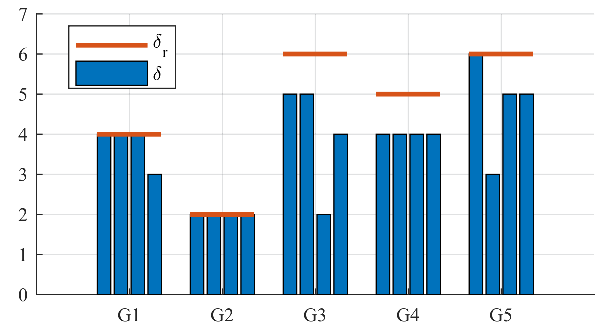

Fig. 4. Value of the security index δ and the robust security index δ_r of Generators 1–5 for different realizations of the system.

the robust indices of all the actuators are increased by the desired values. We show that this problem is an instance of the problem (2), so we can find a suboptimal solution for it in polynomial time with guarantees stated in Lemma 3.

Proposition 7: The problem (9) is an instance of (2).

2) Placement of Protected Sensors: One can also consider the problem of placing protected sensors. One objective could be to increase δ_r to $+\infty$ for as many actuators as possible, which would prevent Attacker 3 from attacking these actuators. Since protected sensors might be expensive, we assume that the operator is limited to $k_{\max}$ sensors.

Let $X_p \subseteq \mathcal{X}$ be a subset of states that we want to measure using the protected sensors and let us define

$$g'_i(X_p) = \min\{|X_p \cap X_i|, 1\}$$

for each u_i. This function returns 1 if there exists a protected sensor measuring a state from X_i. We, then, know from Theorem 3 that $\delta_r(u_i) = +\infty$. Otherwise, $g'_i(X_p) = 0$ holds.

Let $U_p \subseteq \mathcal{U}$ be a subset of actuators for which we want to increase the robust indices to $+\infty$. The problem we want to solve can, then, be formulated as

$$\underset{X_p}{\text{maximize}} \;\; \textstyle\sum_{u_i \in U_p} g'_i(X_p) \quad \text{subject to} \quad |X_p| \leq k_{\max}. \qquad (10)$$

The objective function equals to the number of actuators whose robust indices are equal to $+\infty$ after placing protected sensors at locations X_p. The constraint imposes that no more than $k_{\max}$ protected sensors should be placed. As shown in the following, this problem is an instance of the problem (3). Hence, a suboptimal solution of (10) with $1 - 1/e$ approximation ratio can be obtained in polynomial time (Lemma 4).

Proposition 8: The problem (10) is an instance of (3).

VI. Illustrative Examples

We now discuss the theoretical developments on illustrative numerical examples.

A. Comparison of δ and δ_r

1) Model: Consider the IEEE 14-bus system, shown in Fig. 3. The system is controlled using five generators located at buses 1, 2, 3, 6, and 8. We modeled the system using linearized swing equations where the generators are represented by two states (rotor angle ϕ_i and frequency $\omega_i = \dot{\phi}_i$), and load buses with one state (voltage angle θ_i) [46]. The parameters given in [47] were used. The operator has access to phasor measurement units providing measurements of θ_1, θ_3, θ_5, θ_7, θ_9, θ_{11}, and θ_{13}. We considered the following system realizations:

1) normal operation, as shown in Fig. 3 (Realization 1);
2) power line (Bus 4, Bus 7) switched-off (Realization 2);
3) micro–grid consisting of Bus 3 and Generator 3 detaches from the grid (Realization 3);
4) measurement θ_1 stops being available (Realization 4).

We assumed that every generator and every measurement can be compromised by the attacker, as well as some of the loads [48]. Particularly, the loads at buses $2, 5, 9, 14$ were assumed to have considerable effect to the network, and were modeled as additional actuators.

2) Robustness: We first compare δ and δ_r in terms of robustness. For this purpose, we calculated the values of δ and δ_r for all the generators in the aforementioned four realizations of the system. The results are shown in Fig. 4.

First, the results confirm that δ depends on a realization of the system. Thus, if the operator decides to use δ as a security index, it is not sufficient to consider only one realization. For example, Generator 3 that appears to be the second most secured in Realization 1 becomes one of the two most vulnerable in Realization 3. A less evident observation is that the use of δ can lead to a considerable security allocation cost. Particularly, we see that the minimum value of δ for all the generators is quite similar (except for maybe Generator 4). Therefore, ensuring that every generator has sufficiently large security index δ in every system realization may be very hard and would require a large security investment.

Evidently, the values of δ_r are not dependent on the realization. Therefore, having a small value of δ_r implies that an actuator is vulnerable in any realization. For example, since $\delta_r(G_2) = 2$, Generator 2 can be attacked by Attackers 1 and 2 by compromising only two components in any realization. However, as it can be seen, δ_r is not a tight upper bound on δ. Thus, a large δ_r does not necessarily imply security, which is the main drawback of δ_r. For instance, note that $\delta(G_3) = 2$ in the third realization. Hence, Attacker 1 can conduct a perfectly

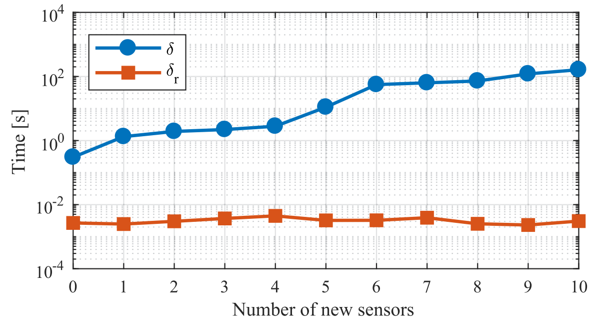

Fig. 5. Computational times required for finding the exact value of δ and δ_r of Generator 4 when the number of sensors vary.

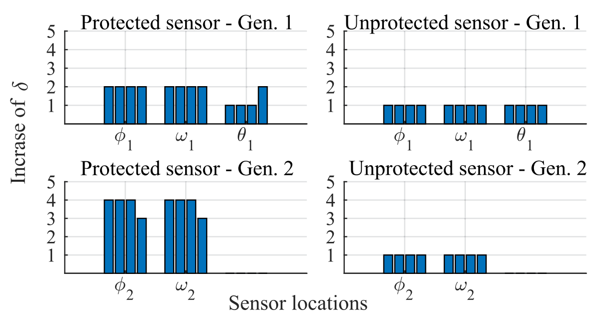

Fig. 6. Increase of the security index δ for Generators 1 and 2.

undetectable attack against Generator 3 in this realization by compromising two components although $\delta_r(G_3) = 6$.

3) Computing δ and δ_r: We now compare the computational efforts needed to calculate δ and δ_r. To calculate δ, we used the brute force search method explained in Section III. To calculate δ_r, we used `maxflow` function that is included in MATLAB R2017. We kept the realization of the system fixed to Realization 1 and varied the number of sensors by placing new sensors at random locations. We, then, measured the times needed to calculate δ and δ_r for Generator 4.

The results are shown in Fig. 5. As expected, the effort for calculating δ grows exponentially with the number of newly added sensors. Furthermore, note that this effort scales with the number of realizations for which we want to calculate δ. The time needed for calculating δ_r was almost not affected by placing this relatively small number of sensors and remained below 0.01 s in all the cases. Additionally, δ_r is calculated only once, since it has the same value in any realization.

4) Increasing δ and δ_r: We now investigate if by increasing δ_r we also increase δ. We focus on Generators 1 and 2, since these generators have the lowest values of δ_r. Using Theorem 3, we obtained that suitable locations for placing additional sensors are $X_1 = \{\phi_1, \omega_1, \theta_1\}$ for Generator 1 and $X_2 = \{\phi_2, \omega_2\}$ for Generator 2.

We first investigated how the placement of one protected sensor at the locations from X_1 influences δ. While placing the protected sensor at these locations increases $\delta_r(G_1)$ to $+\infty$, it can be seen from Fig. 6 that $\delta(G_1)$ did not increase to $+\infty$ in any of the four realizations we considered. Yet, the increase of $\delta(G_1)$ for more than one was achieved in majority of the

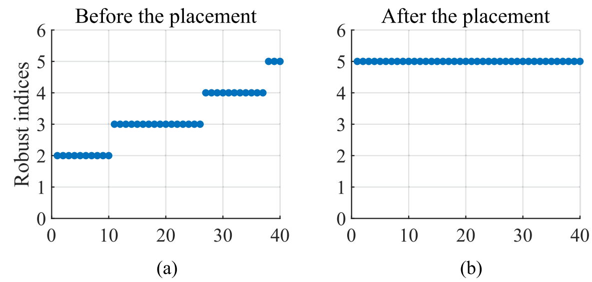

Fig. 7. Robust security indices before and after the sensor placement.

cases, which is impossible to achieve by placing an unprotected sensor (Proposition 2). The experiment was also conducted for Generator 2. Similarly, $\delta(G_2)$ did not increase to $+\infty$ in any of the four realizations. However, the placement of one protected sensor led to increase of $\delta(G_2)$ by at least three for all the locations from X_2 and all the realizations.

We also considered placing one unprotected sensors at locations from X_1, which increases $\delta_r(G_1)$ by one. Interestingly, from Fig. 6, the placement of one unprotected sensor at any of the locations from X_1 led to increase of $\delta(G_1)$ in all the realizations. The same holds for X_2 and $\delta(G_2)$.

Overall, the experiment illustrates that by increasing δ_r, we can also indirectly increase δ. However, from the placement of protected sensors, we see that we definitely do not achieve the same level of improvement. This again illustrates that protecting the system against advanced Attacker 1 may require much more resources than protecting it against less advanced attackers such as Attacker 3.

B. Increasing δ_r in Large-Scale Networked Control Systems

We now consider the problem of improving δ_r in the IEEE 2383-bus system. This large-scale system has 3037 states and 327 generators. We modeled the system in the same way as the IEEE 14-bus system, selected randomly 40% of the states to be measurable, and 10% of the load buses to be attackable. We, then, calculated the robust indices of all the generators and plotted the smallest 40 robust security indices in Fig. 7(a). We emphasize that it took only 114.03 s to calculate all the robust indices, which confirms that these indices can be calculated efficiently in large-scale systems. As one can see, there are 37 generators with the robust indices equal to 2, 3, or 4, which makes these generators vulnerable in any realization of the system. Therefore, we also considered the problem of placing unprotected sensors such as to make all the robust indices to be at least equal to 5. For this purpose, we formed and solved the problem (9), which took only 0.5654 s. As one can see from Fig. 7(b), the robust indices were successfully increased after the placement.

C. Properties of Full and Limited Model Knowledge Attackers

We now illustrate the limitations of the full and limited model knowledge attackers considered in the paper. For this purpose, we consider the system of two autonomous vehicles shown in

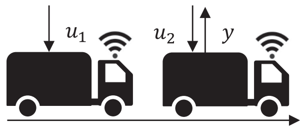

Fig. 8. Platoon consisting of two autonomous vehicles. The vehicles can be controlled by the operator through the signals u_1 and u_2. The operator also knows the position of the second vehicle y.

Fig. 8. Each vehicle is modeled by a single state representing its position relative to some moving reference frame. The operator can control both vehicles through the signals u_1 and u_2 and knows the position of the second vehicle $y = x_2$. The operator's goal is to keep the distance between the vehicles equal to 10. To study this formation control problem, we use the model from [8]

$$x(k+1) = \begin{bmatrix} 1-2\alpha_1 & \alpha_1 \\ \alpha_2 & 1-2\alpha_2 \end{bmatrix} x(k) + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} u(k)$$

$$y(k) = \begin{bmatrix} 0 & 1 \end{bmatrix} x(k)$$

where $\alpha_1 = \alpha_2 = 0.1$. We initially assume that $x(0) = [0 \quad 10]^T$ and $u(k) = [-1 \quad 2]^T$ for any $k \in \mathbb{Z}_{\geq 0}$, so that desired behavior of the platoon is achieved prior to attacks.

We consider Attackers 1 and 2.[2] Both of the attackers control u_1 and y, and have the goal to disrupt the platoon formation without the operator noticing. In the following, we discuss in which situations the attackers can achieve this objective. By Δy_F (resp. Δy_L), we denote the difference between the measurement expected in the normal operation and the received measurement in the case of Attacker 1 (resp. Attacker 2). Attacker 1 (resp. Attacker 2) remains perfectly undetectable if $\Delta y_F \equiv 0$ (resp. $\Delta y_L \equiv 0$) holds.

Case 1: The first case illustrates that both of the attackers can conduct a perfectly undetectable attack once the system is in a steady state. Attacker 1 applies the following signals:

$$a_1(k) = -k$$
$$a_3(k+2) = 1.6a_3(k+1) - 0.63a_3(k) - 0.1a_1(k) \quad (11)$$

which is according to the strategy introduced in the proof of Proposition 1. Attacker 2 applies the signals

$$a_1(k) = -k, \quad a_3(k) = -x_2(k) + y(0) \quad (12)$$

which is according to the strategy introduced in the proof of Proposition 5. As we can see from Fig. 9, Case 1, both of the attackers remain perfectly undetectable.

Case 2: This case illustrates the sensitivity of Attacker 1 with respect to modeling errors. Assume that Attacker 1 believes that $\alpha_2' = 0.11$. He/she, then, applies the signals

$$a_1(k) = -k$$
$$a_3(k+2) = 1.58a_3(k+1) - 0.613a_3(k) - 0.11a_1(k).$$

[2]The properties of Attacker 2 we outline next are the same as for Attackers 3, which is the reason why we do not explicitly consider Attackers 3.

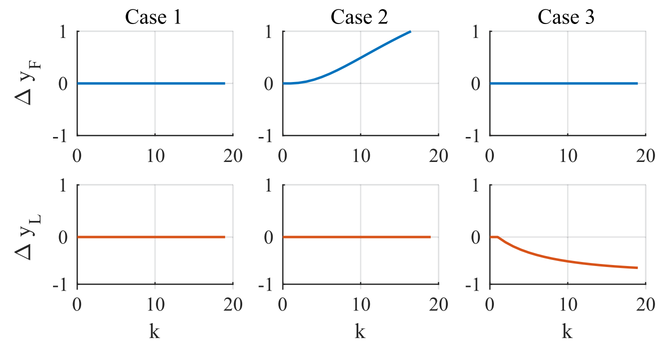

Fig. 9. Difference of the expected and attacked sensor measurement in three different cases.

Attacker 2 applies the same signals as in the previous case. From Fig. 9, Case 2, we can see that Attacker 1 is revealed, while Attacker 2 remains undetected. Generally, Attacker 2 can also be vulnerable to modeling errors, since he/she may require precise local model knowledge to construct the strategy. However, the fact that this attacker uses only a fraction of the model (in this case none), lowers his/her chances to become detected because of modeling errors.

Case 3: Finally, assume the scenario where the operator increases the control signal u_2 by 0.1 at $k = 2$ and the attackers apply the signals (11) and (12). From Fig. 9, Case 3, we see that Attacker 2 is revealed. This illustrates that the steady-state assumption is generally required for Attacker 2 to remain perfectly undetectable. Namely, Attacker 2 does not know neither u_2 nor the equation for x_2. Hence, when y starts changing, he/she cannot distinguish if this is because of the attack or a change in u_2. We also see that Attacker 1 remains undetected. The reason is that the signals (11) can be calculated prior to the attack and implemented in a feedforward manner, which makes the attack decoupled from $x(0)$ and u.

VII. Conclusion and Future Work

We introduced the actuator security indices δ and δ_r that can be used for localizing vulnerable actuators within the system and development of defense strategies. A method for computing δ was derived and it was shown that δ can potentially be increased by placing additional sensors. We, then, showed that δ may not be an appropriate index for large-scale systems since it is NP-hard to calculate, sensitive to system variations, and based on the assumption that the attacker knows the entire system model. In contrast, the robust index δ_r can be calculated efficiently, can characterize actuators vulnerable in any realization, and can be related to both the full and limited model knowledge attackers. The drawback of δ_r is that it cannot be used to detect actuators vulnerable in a particular system realization. Additionally, two sensor placement problems for increasing δ_r were proposed, and it was shown that suboptimal solutions of these problems with performance guarantees can be calculated efficiently.

The future work may go into the following directions. First, besides perfect undetectability, there exist other ways to define undetectability. Hence, we plan to investigate if novel types of indices can be formulated based on these definitions. Second, the sensor placement strategies we developed do not take the index δ into consideration. The future work will investigate if it is possible to increase δ and δ_r simultaneously. Finally, it would be interesting to take the probability that a realization of the system will appear into account. The attacker may, then, want to gather resources such as to conduct a successful attack with sufficiently high probability, which would require us to derive new security indices.

Appendix

A. Proofs of Section III

Proof of Proposition 1: We first introduce an auxiliary lemma.

Lemma 5 (see [8, Th. 1][9, Th. 7]): A perfectly undetectable attack conducted using components $I_a \subseteq I$ exists if and only if nrank $G^{(I_a)} < |I_a|$.

Proof of Proposition 1: (⇒) Let $\mathcal{A}$ be the $\mathcal{Z}$-transform of a and assume there exists a perfectly undetectable attack $\mathcal{A}$ with $\mathcal{A}_i \neq 0$. We split the proof into two cases.

Case 1: nrank $G^{(I_a \setminus i)} = |I_a| - 1$. Since undetectable attacks are possible, then nrank $G^{(I_a)} < |I_a|$ (Lemma 5). In addition

$$\text{nrank } G^{(I_a)} \geq \text{nrank } G^{(I_a \setminus i)} = |I_a| - 1$$

which implies nrank $G^{(I_a)} = |I_a| - 1$. Thus, (5) holds.

Case 2: nrank $G^{(I_a \setminus i)} = r < |I_a| - 1$. Let $z \in \mathbb{C}$ be such that rank $G^{(I_a \setminus i)}(z) = r$ and let I_b be a set that contains indices of any r linearly independent columns of $G^{(I_a \setminus i)}(z)$. Since nrank $G^{(I_b)} \leq |I_b| = r$ (the number of columns of $G^{(I_b)}$ is $|I_b|$) and nrank $G^{(I_b)} \geq$ rank $G^{(I_b)}(z) = r$, it follows that

$$\text{nrank } G^{(I_a \setminus i)} = \text{nrank } G^{(I_b)} = r. \tag{13}$$

Next, note that nrank $[G^{(I_b)} G^{(j)}] = r$ has to hold for any $j \in I_a \setminus i$ (nrank $G^{(I_a \setminus i)}$ would be greater than r otherwise). Hence, we can find rational matrices P and $Q \neq 0$ that satisfy $G^{(I_b)}P + G^{(j)}Q = 0$ [49, p. 31], which implies that the columns of $G^{(I_b)}$ span all the columns of $G^{(I_a \setminus i)}$. Hence, we can find $\mathcal{A}'$ for which $G^{(I_a \setminus i)}\mathcal{A}^{(I_a \setminus i)} = G^{(I_b)}\mathcal{A}'$, where $\mathcal{A}^{(I_a \setminus i)}$ is the vector consisting of the elements of $\mathcal{A}$ with indices from $I_a \setminus i$. From the latter and $G\mathcal{A} = 0$, we have

$$G\mathcal{A} = G^{(I_a \setminus i)}\mathcal{A}^{(I_a \setminus i)} + G^{(i)}\mathcal{A}_i = G^{(I_b)}\mathcal{A}' + G^{(i)}\mathcal{A}_i = 0.$$

This implies that $[\mathcal{A}'^T \ \mathcal{A}_i]^T$ is a perfectly undetectable attack against $[G^{(I_b)} \ G^{(i)}]$ with $\mathcal{A}_i \neq 0$. From this fact and nrank $G^{(I_b)} = |I_b|$, it follows from Case 1 that the condition (5) holds for the set of components $I_b \cup i$. Thus, we have

$$\text{nrank } [G^{(I_b)} \ G^{(i)}] \overset{\text{Case1}}{=} \text{nrank } G^{(I_b)} \overset{(13)}{=} \text{nrank } G^{(I_a \setminus i)}. \tag{14}$$

Since $G^{(I_b)}$ spans the columns of $G^{(I_a \setminus i)}$, we have

$$\text{nrank } [G^{(I_b)} \ G^{(i)}] = \text{nrank } [G^{(I_a \setminus i)} \ G^{(i)}] = \text{nrank } G^{(I_a)}. \tag{15}$$

From (14) and (15), we conclude that (5) holds.

(⇐) If (5) holds, then there exist real rational functions P and $Q \neq 0$, such that $G^{(I_a \setminus i)}P + G^{(i)}Q = 0$. Thus, an arbitrary attack signal $\mathcal{A}_i$ can be masked by applying $\mathcal{A}^{(I_a \setminus i)} = P\mathcal{A}_i/Q$ on the remaining attacked components. ∎

Proof of Proposition 2: By placing a new sensor, we introduce additional constraints to Problem 1. These constraints shrink the set of feasible points. Thus, $\delta'(u_i) < \delta(u_i)$ cannot hold. If a new sensor is not protected, the attacker can compromise it. This can be interpreted as removing the aforementioned constraints from the problem. Hence, $\delta'(u_i)$ is at most by one larger than $\delta(u_i)$ once a new sensor is unprotected. By adding a new actuator, the number of decision variables of Problem 1 increases and the constraints remain the same. Therefore, we conclude that $\delta'(u_i) \leq \delta(u_i)$ holds. ∎

Proof of Theorem 1: To prove NP-hardness of Problem 1, it suffices to show that every instance of an NP-hard problem can be mapped into Problem 1. For this purpose, we use the NP-hard sparse recovery problem [50]

$$\underset{d}{\text{minimize}} \ \|d\|_0 \quad \text{subject to} \quad Fd = z \tag{16}$$

where $F \in \mathbb{R}^{p \times m}$ and $z \in \mathbb{R}^p$ are given.

Let F and z be arbitrary selected. Set $A = 0_{m \times m}$, $B = I_m$, $C = [-z \ F]$, $D_a = 0_{p \times m}$, and $u_i = u_1$. Then, $x(k+1) = a(k)$ and $y(k) = Cx(k)$, so Problem 1 becomes

$$\underset{a}{\text{minimize}} \ \|a\|_0 \quad \text{subject to} \quad Ca(k) = 0, \quad a_1 \not\equiv 0. \tag{17}$$

To solve (17) for all k, it suffices to solve it for a single k. Thus, (17) reduces to

$$\underset{a(0)}{\text{minimize}} \ \|a(0)\|_0 \quad \text{subject to} \quad Ca(0) = 0, a_1(0) = 1$$

where the substitution of $a_1(0) \neq 0$ with $a_1(0) = 1$ is without loss of generality. Let $a(0) = [1 \ d^T]^T$. Then, minimizing $\|a(0)\|_0$ is equivalent to minimizing $\|d\|_0$, which is the objective function of (16). Moreover, we also have that

$$Ca(0) = [-z \ F]\begin{bmatrix} 1 \\ d \end{bmatrix} = -z + Fd.$$

Thus, the constraint $Ca(0) = 0$ becomes the constraint from (16). Hence, every instance of the NP-hard problem (16) can be mapped into Problem 1, which concludes the proof. ∎

B. Proofs of Section V

Proof of Theorem 2: (⇐) Let $X_a \cup Y_a$ be a vertex separator of u_i and t in $\mathcal{G}_t$. To prove the claim, we introduce an attack strategy that only uses the components U_a and Y_a. We, then, prove that this strategy is actively using u_i and it is perfectly undetectable for any $(A, B, C) \in \mathcal{R}$.

For the actuator u_i, the attacker injects any signal $a_i \not\equiv 0$. This ensures that u_i is actively used in the attack. For any other attacked actuator $u_j \in U_a \setminus u_i$, the attack is given by

$$a_j(k) = -\frac{A(p,:)}{B(p,j)}x(k) \tag{18}$$

where $A(p,:)$ is the row of A corresponding to the actuator u_j and $B(p,j)$ is the nonzero element of B multiplying u_j ($B(p,j) \neq 0$ in every realization due to Assumption 2). For any attacked sensor $y_l \in Y_a$, the attack is given by

$$a_{n_u+l}(k) = -C(l,:)x(k) \tag{19}$$

where $C(l,:)$ represents the row of C corresponding to y_l. For the attacker with the full model knowledge, this strategy can be constructed for any realization. Namely, he/she knows the values for $A(p,:), B(p,j), C(l,:)$, and can predict the value of $x(k)$ for any $k \in \mathbb{Z}_{\geq 0}$ based on the model and the attack signals. We now prove that this strategy is perfectly undetectable, that is, $y \equiv 0$.

Consider first the attacked sensors. For any $y_l \in Y_a$ and $k \in \mathbb{Z}_{\geq 0}$, we have $y_l(k) = C(l,:)x(k) + a_{n_u+l}(k) \overset{(19)}{=} 0$. Thus, the attacked measurements are equal to 0.

Consider now the nonattacked measurements of the states X_a. Let $x_p \in X_a$ and let $u_j \in U_a \setminus u_i$ be adjacent to x_p. Then, $x_p(k+1) = A(p,:)x(k) + B(p,j)a_j(k) \overset{(18)}{=} 0$. Thus, the nonattacked measurements of the states from X_a are equal to 0. Let now X_b be the set of all the states for which there exists a directed path from u_i that does not contain the states from X_a. These states cannot be measured using the nonattacked sensors. That would imply that there exists a directed path between u_i and t not intersected by $X_a \cup Y_a$, which is in contradiction with the assumption that $X_a \cup Y_a$ is a vertex separator of u_i and t. Finally, let $X_c = \mathcal{X} \setminus (X_b \cup X_a)$. Note that the directed edges $(x_b, x_c), x_b \in X_b, x_c \in X_c$ cannot exist. That would imply that there exists a directed path from u_i to x_c that does not contain the states from X_a, so x_c would belong to X_b. Thus, the states from X_c cannot be directly influenced by the states from X_b. Since $x(0) = 0$, $u \equiv 0$, and the states X_a remain equal to 0, the states X_c also remain equal to 0 during the attack. Thus, the nonattacked measurements of the states X_c remain 0. With this, we prove that all of the nonattacked measurements are equal to 0, so the attack strategy is perfectly undetectable.

($\Rightarrow$) The proof is by contradiction. If $X_a \cup Y_a$ is not a vertex separator of u_i and t in $\mathcal{G}_t$, then there exists a simple directed path $u_i, x_{i_0}, \ldots, x_{i_n}, y_l, t$ (Path 1) not intersected by $X_a \cup Y_a$. We show that this implies existence of at least one realization $(A, B, C) \in \mathcal{R}$ in which perfectly undetectable attacks against u_i cannot be conducted.

Particularly, assume the following feasible realization of matrices A and C. For x_{i_0} from Path 1, $A(i_0,:) = 0$. This ensures that x_{i_0} cannot be influenced by other states. For any other state x_{i_k} from Path 1, $A(i_k, j) \neq 0$ (resp. $A(i_k, j) = 0$) if $j = i_{k-1}$ (resp. $j \neq i_{k-1}$). This guarantees that the only state that influences x_{i_k} is $x_{i_{k-1}}$. Finally, let $C(l, i_n) \neq 0$, which ensures that $y_l(k) \neq 0$ once $x_{i_n}(k) \neq 0$.

Let $a_i \not\equiv 0$ be an arbitrary attack signal against u_i, and let k_0 be the first time instant for which $a_i(k_0) \neq 0$. Since a_i is the only attack signal that can directly influence x_{i_0} (see Assumption 2) and $A(i_0,:) = 0$, we have

$$x_{i_0}(k_0+1) = B(i_0, i)a_i(k_0) \neq 0.$$

Note that the only state that influences x_{i_1} is x_{i_0} and x_{i_1} cannot be directly influenced by other attacked actuators ($x_{i_1} \notin X_a$). Hence, we have

$$x_{i_1}(k_0+2) = A(i_1, i_0)x_{i_0}(k_0+1) \neq 0.$$

By applying the similar reasoning to all the remaining states from Path 1, it can be shown that $x_{i_n}(k_0+n+1) \neq 0$. From $C(l, i_n) \neq 0$, we have $y_l(k_0+n+1) \neq 0$. Thus, the attack is revealed. Since a_i was arbitrary selected, there exists no perfectly undetectable attacks with u_i actively used in this realization, which establishes the claim. ■

Proof of Proposition 3: Assume that (U_a, Y_a) is a solution of the problem (8) and let $X_a \cup Y_a$ be the corresponding vertex separator. Let $E_c \subseteq \mathcal{E}_i$ be constructed as follows. For each $x_k \in X_a$, we add $(x_{k_{\text{in}}}, x_{k_{\text{out}}})$ to E_c. For each $y_j \in Y_a$ with $(x_k, y_j) \in \mathcal{E}_{xy}$, we add $(x_{k_{\text{out}}}, t)$ (resp. (x_k, t)) to E_c if x_k is Type 1 (resp. Type 2). If several sensors measure x_k, then all of them must belong to Y_a. Otherwise, there would exist a path from u_i to t not intersected by $X_a \cup Y_a$ or y_j would not be a part of an optimal solution. From the construction of $\mathcal{G}_i$, the edges added to E_c have the cost $\delta_c = |U_a \setminus i| + |Y_a| = \delta_r(u_i) - 1$. We now show that E_c is an edge separator of u_i and t in $\mathcal{G}_i$ (Claim 1) of the minimum cost (Claim 2). This implies that $\delta_r(u_i) = \delta_c + 1 = \delta^* + 1$ holds.

Claim 1. If E_c is not an edge separator of u_i and t, then there exists a simple directed path $u_i, x_{j_1}, \ldots, x_{j_n}, t$ (Path 1) in $\mathcal{G}_i$ not intersected by E_c. By the construction of $\mathcal{G}_i$ that implies existence of a simple directed path $u_i, x_{k_1}, \ldots, x_{k_m}, y_l, t$ (Path 2) in $\mathcal{G}_t$ is obtained from Path 1 by replacing every pair $x_{p_{\text{in}}}, x_{p_{\text{out}}}$ that corresponds to x_p of Type 1 by x_p and by inserting a measurement y_l of x_{k_m}. Path 2 has to be intersected by $X_a \cup Y_a$, so there either exists $x_p \in X_a$ that belongs to Path 2 or $y_l \in Y_a$. Then, either $(x_{p_{\text{in}}}, x_{p_{\text{out}}})$ or (x_{j_n}, t) belongs to E_c. This contradicts existence of Path 1, so Claim 1 holds.

Claim 2. Assume there exists an edge separator E'_c with a cost $\delta' < \delta_c$. Let U'_a and Y'_a be constructed as follows. For each $(x_{k_{\text{in}}}, x_{k_{\text{out}}})$ from E'_c, we add u_j to U'_a, where u_j is adjacent to x_k. For each edge $(x_{p_{\text{out}}}, t)$ or (x_p, t) from E'_c, we add all the measurements of x_p to Y'_a. All of these measurements must be unprotected (otherwise $\delta' = +\infty > \delta_c$). Finally, we add u_i to U'_a. Note that E'_c cannot contain edges of other types, because their weight is $+\infty$, which would imply $\delta' > \delta_c$.

We first prove that (U'_a, Y'_a) is a feasible point of (8). Assume that is not the case. Then, there exists a simple directed path $u_i, x_{k_1}, \ldots, x_{k_m}, y_l, t$ (Path 1') in $\mathcal{G}_t$ consisting of the states that are not adjacent to $U'_a \setminus u_i$ and $y_l \notin Y'_a$. We can, then, construct Path 2' in $\mathcal{G}_i$ by replacing each node x_p of Type 1 from Path 1' by $x_{p_{\text{in}}}, x_{p_{\text{out}}}$ and removing y_l from Path 1'. By the construction of U'_a, Y'_a, and $\mathcal{G}_i$, Path 2' cannot be intersected by E'_c. This would contradict the assumption that E'_c is an edge separator, so (U'_a, Y'_a) has to be a feasible point of the problem (8). Yet, (U_a, Y_a) is, then, not a solution of the problem (8) because $|U'_a| + |Y'_a| = \delta' + 1 < |U_a| + |Y_a| = \delta_c + 1$. Thus, E'_c cannot exist and Claim 2 holds.

If $\delta_r(u_i) = +\infty$, then there exists a simple directed path $u_i, x_{j_1}, \ldots, x_{j_n}, y_l, t$ in $\mathcal{G}_t$ that consists of u_i, Type 2 states, a protected measurement, and t. Then, the path $u_i, x_{j_1}, \ldots, x_{j_n}, t$ exists in $\mathcal{G}_i$ and the weights of the edges from this path are equal to $+\infty$. Since any edge separator needs to cut this path, we conclude that $\delta^* = +\infty$ holds. ■

Proof of Proposition 4: Case $\delta_r(u_i) < +\infty$: Let (U_a, Y_a) be a solution of the problem (8). The attacker can, then, conduct a perfectly undetectable attack against u_i in any realization using U_a and Y_a, so $\delta(u_i) \leq |U_a| + |Y_a| = \delta_r(u_i)$ holds.

Case $\delta_r(u_i) = +\infty$: The proof is by contradiction. Assume that $\delta(u_i) \neq +\infty$ in every realization from $\mathcal{R}$, and let $U_a = \mathcal{U}$ and Y_a be the set of all unprotected sensors. Since $\delta(u_i) \neq +\infty$, we conclude that there exists a solution of Problem 1 in any realization from $\mathcal{R}$. However, if the attacker can conduct a perfectly undetectable attack against u_i in a particular realization with some set of components, then he/she can do it with U_a and Y_a as well. It, then, follows that (U_a, Y_a) is a feasible point of the problem (8), which is impossible since $\delta_r(u_i) = +\infty$. Therefore, $\delta(u_i) = +\infty$ has to hold for at least one realization from $\mathcal{R}$. ■

Proof of Proposition 5: ($\Rightarrow$) The proof is by contradiction. If $X_a \cup Y_a$ is not a vertex separator of u_i and t in $\mathcal{G}_t$, we know from the proof of Theorem 2 that we can find at least one realization in which it is impossible to conduct a perfectly undetectable attack against u_i. Thus, $X_a \cup Y_a$ has to be a vertex separator of u_i and t.

($\Leftarrow$) If $X_a \cup Y_a$ is a vertex separator of u_i and t, Attacker 2 can conduct a perfectly undetectable attack against u_i using the strategy similar to the one in the proof of Theorem 2. For actuator u_i, the attacker injects an arbitrary signal $a_i \not\equiv 0$. If $u_j \in U_a \setminus u_i$ with $(u_j, x_p) \in \mathcal{E}_{ux}$, the attack is given by $a_j(k) = -A(p,:)x(k)/B(p,j)$, where $A(p,:)$ is the row of A corresponding to attacked actuator u_j, and $B(p,j)$ is the nonzero element of B multiplying u_j. For $y_l \in Y_a$, the attacker selects $a_{l+n_u}(k)$ to maintain $y_l(k) = 0$ for any $k \in \mathbb{Z}_{\geq 0}$.

Attacker 2 can construct this attack. First, Attacker 2 knows the values for $A(p,:), B(p,:)$ that correspond to actuators $u_j \in \mathcal{U} \setminus u_i$. Second, the attacker can construct $A(p,:)x(k)$, since he/she knows the values of in-neighbors of x_p (the elements of $A(p,:)$ that correspond to other states are equal to 0). Third, Attacker 2 can also set the signals of attacked sensors and actuators to an arbitrary value, so he/she can maintain $y_l(k) = 0$ for any $k \in \mathbb{Z}_{\geq 0}$. The proof that $y \equiv 0$ can, then, be found in the proof of Theorem 2. ■

Proof of Proposition 6: The proof is by contradiction. Assume that Y_a is not a vertex separator of u_i and t in $\mathcal{G}_t$. Then, there exists a sensor y_j not compromised by Attacker 3 and a directed path from u_i to y_j. We, then, know from the proof of Theorem 2 that there exists at least one realization in which any attack against u_i triggers y_j. Since Attacker 3 knows only $[A]$, $[B]$, $[C]$, he/she does not know if an attack against u_i would be visible in y_j. Thus, Attacker 3 needs to attack y_j to ensure being perfectly undetectable. Therefore, Y_a has to form a vertex separator of u_i and t. Since $\delta_r(u_i) - 1$ is the size of the minimum vertex separator of u_i and t in $\mathcal{G}_t$ (we subtract 1 from $\delta_r(u_i)$ to exclude u_i), we have that $|Y_a| \geq \delta_r(u_i) - 1$ holds. If $\delta_r(u_i) = +\infty$, then there exists a path between u_i and a protected sensor. Hence, Y_a cannot be a vertex separator of u_i and t. Attacker 3, then, cannot ensure that an attack against u_i remains perfectly undetectable, because he/she does not know if the protected sensor would be triggered. ■

Proof of Theorem 3: If we place a sensor y_j to measure a state from X_i, then we introduce at least one directed path from u_i to t that does not contain states adjacent to $\mathcal{U} \setminus u_i$. Thus, the only way to eliminate this path is by adding y_j to a vertex separator. If y_j is protected, then that is not possible. Hence, $\delta_r'(u_i) = +\infty$ holds. Otherwise, the attacker has to attack y_j, in which case $\delta_r'(u_i) = \delta_r(u_i) + 1$ holds.

We now show that if every state directly controlled by an actuator is also directly measured by a sensor, then the only way to improve $\delta_r(u_i)$ is by placing sensors within X_i. Let (U_a, Y_a) be a solution of (8) for u_i. We first form another solution (U_a', Y_a'). The set Y_a' is formed by removing from Y_a any y_k, which measures $x_l \in \mathcal{X}$ that is adjacent to $u_m \in \mathcal{U} \setminus u_i$. As a substitute of y_k, we add u_m to U_a'. We, then, add all the actuators U_a to U_a'. This ensures that for all the states that are both directly controlled by an actuator and measured by a sensor, we always select an actuator to belong to a solution of (8) rather than a sensor.

Let X_a' be defined as in (7) based on U_a' and let a new sensor y_j be placed on a location $x_l \notin X_i$. If there are no directed paths from u_i to x_l, then $X_a' \cup Y_a'$ is still a vertex separator of u_i and t and $\delta_r(u_i)$ is not increased. Assume now that there exists a simple directed path $u_i, \ldots, x_l, y_j, t$ (Path 1). Since $x_l \notin X_i$, there has to exist a state x_p from Path 1 adjacent to an actuator from $\mathcal{U} \setminus u_i$. Suppose that $X_a' \cup Y_a'$ is not a vertex separator of u_i and x_p and that $x_p \notin X_a'$. Since every state adjacent to an actuator is adjacent to a sensor, it follows that there exists a directed path between u_i and t passing through x_p that is not intersected by $X_a' \cup Y_a'$. This is impossible, since (U_a', Y_a') is a solution of (8). Therefore, $X_a' \cup Y_a'$ has to be a vertex separator of u_i and x_p or $x_p \in X_a'$. This implies that $X_a' \cup Y_a'$ intersects Path 1. The same holds for any other path between u_i and y_j. Hence, $\delta_r(u_i)$ cannot be increased by measuring states outside X_i. ■

Proof of Proposition 7: It suffices to show that $\sum_{u_i \in \mathcal{U}} g_i(Y_p)$ is submodular, nondecreasing, and integer-valued. First, $w_{ji} = |x_{y_j} \cap X_i|$ equals to 0 or 1. Thus, $f_i(Y_p) = \sum_{y_j \in Y_p} w_{ji}$ is a linear function, so it is submodular [33, Sec. 2] and nondecreasing (sum of non-negative numbers). Since $g_i(Y_p) = \min\{f_i(Y_p), k_i\}$, it follows from Lemma 2 that g_i is submodular and nondecreasing. Function g_i is also integer valued, since f_i and k_i are integer valued. From the previous discussion and Lemma 1, it follows that $\sum_{u_i \in \mathcal{U}} g_i(Y_p)$ is submodular, nondecreasing, and integer valued. Hence, the claim of the proposition holds. ■

Proof of Proposition 8: The function g_i' is known to be submodular [33, Sec. 2]. Additionally, g_i' is a nondecreasing function, since $|X_p \cap X_i|$ is nondecreasing in X_p. We, then, have from Lemma 1 that $\sum_{u_i \in U_p} g_i'(X_p)$ is submodular and nondecreasing. Thus, $\sum_{u_i \in U_p} g_i'(X_p)$ has the same properties as the objective function of (3), which concludes the proof. ■

References

[1] F. L. Cortesi, T. H. Summers, and J. Lygeros, “Submodularity of energy related controllability metrics,” in *Proc. IEEE Conf. Decision Control*, Dec. 2014, pp. 2883–2888.

[2] F. Pasqualetti, S. Zampieri, and F. Bullo, “Controllability metrics, limitations and algorithms for complex networks,” *IEEE Trans. Control Netw. Syst.*, vol. 1, no. 1, pp. 40–52, Mar. 2014.

[3] V. Tzoumas, M. A. Rahimian, G. J. Pappas, and A. Jadbabaie, “Minimal actuator placement with bounds on control effort,” *IEEE Trans. Control Netw. Syst.*, vol. 3, no. 1, pp. 67–78, Mar. 2016.

[4] A. Clark, L. Bushnell, and R. Poovendran, "On leader selection for performance and controllability in multi-agent systems," in *Proc. IEEE Conf. Decision Control*, Dec. 2012, pp. 86–93.

[5] J. Slay and M. Miller, "Lessons learned from the Maroochy water breach", in *Critical Infrastructure Protection*. Berlin, Germany: Springer, 2008.

[6] D. Kushner, "The real story of STUXNET," *IEEE Spectr.*, vol. 50, no. 3, pp. 48–53, Mar. 2013.

[7] R. Lee, M. Assante, and T. Conway, "Analysis of the cyber attack on the Ukrainian power grid," Electricity Information Sharing and Analysis Center, Washington, D.C., USA, 2016.

[8] H. Cam, P. Mouallem, Y. Mo, B. Sinopoli, and B. Nkrumah, "Modeling impact of attacks, recovery, and attackability conditions for situational awareness," in *Proc. IEEE Int. Inter-Disciplinary Conf. Cogn. Method Situation Awareness Decis. Support*, 2014, pp. 181–187.

[9] S. Weerakkody, X. Liu, S. H. Son, and B. Sinopoli, "A graph-theoretic characterization of perfect attackability for secure design of distributed control systems," *IEEE Trans. Control Netw. Syst.*, vol. 4, no. 1, pp. 60–70, Mar. 2017.

[10] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *Proc. 28th Int. Conf. Distrib. Comput. Syst. Workshops*, 2008, pp. 495–500.

[11] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Trans. Autom. Control*, vol. 59, no. 6, pp. 1454–1467, Jun. 2014.

[12] Y. Mo and B. Sinopoli, "On the performance degradation of cyber-physical systems under stealthy integrity attacks," *IEEE Trans. Autom. Control*, vol. 61, no. 9, pp. 2618–2624, Sep. 2016.

[13] F. Pasqualetti, F. Dorfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Trans Autom. Control*, vol. 58, no. 11, pp. 2715–2729, Nov. 2013.

[14] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents Part I: Attacking the network," in *Proc. Am. Control Conf.*, 2008, pp. 1350–1355.

[15] F. Pasqualetti, A. Bicchi, and F. Bullo, "Consensus computation in unreliable networks: A system theoretic approach," *IEEE Trans. Autom. Control*, vol. 57, no. 1, pp. 90–104, Jan. 2012.

[16] Y. Liu, P. Ning, and M. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Inf. Syst. Security*, vol. 14, no. 1, pp. 13:1–13:33, 2011.

[17] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[18] Y. Z. Lun, A. DInnocenzo, F. Smarra, I. Malavolta, and M. D. D. Benedetto, "State of the art of cyber-physical systems security: An automatic control perspective," *J. Syst. Softw.*, vol. 149, pp. 174–216, 2019.

[19] J. Giraldo, E. Sarkar, A. Cardenas, M. Maniatakos, and M. Kantarcioglu, "Security and privacy in cyber-physical systems: A survey of surveys," *IEEE Design Test*, vol. 34, no. 4, pp. 7–17, Aug. 2017.

[20] H. Sandberg, S. Amin, and K. H. Johansson, "Cyberphysical security in networked control systems: An introduction to the issue," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 20–23, Feb. 2015.

[21] H. Sandberg, A. Teixeira, and K. Johansson, "On security indices for state estimators in power networks," in *Proc. 1st Workshop Secure Control Syst.*, 2010.

[22] O. Vuković, K. Sou, G. Dan, and H. Sandberg, "Network-aware mitigation of data integrity attacks on power system state estimation," *IEEE J. Sel. Area Commun.*, vol. 30, no. 6, pp. 1108–1118, Jul. 2012.

[23] J. M. Hendrickx, K. H. Johansson, R. M. Jungers, H. Sandberg, and K. C. Sou, "Efficient computations of a security index for false data attacks in power networks," *IEEE Trans. Autom. Control*, vol. 59, no. 12, pp. 3194–3208, Dec. 2014.

[24] K. C. Sou, H. Sandberg, and K. H. Johansson, "Electric power network security analysis via minimum cut relaxation," in *Proc. IEEE Conf. Decis. Control Eur. Control Conf.*, 2011, pp. 4054–4059.

[25] K. C. Sou, H. Sandberg, and K. H. Johansson, "Computing critical k-tuples in power networks," *IEEE Trans. Power Syst.*, vol. 27, no. 3, pp. 1511–1520, Aug. 2012.

[26] O. Kosut, "Max-flow min-cut for power system security index computation," in *Proc. IEEE 8th Sens. Array Multichannel Signal Process. Workshop*, 2014, pp. 61–64.

[27] Y. Yamaguchi, A. Ogawa, A. Takeda, and S. Iwata, "Cyber security analysis of power networks by hypergraph cut algorithms," *IEEE Trans. Smart Grid*, vol. 6, no. 5, pp. 2189–2199, Feb. 2015.

[28] M. S. Chong and M. Kuijper, "Characterising the vulnerability of linear control systems under sensor attacks using a system's security index," in *Proc. IEEE Conf. Decis. Control*, 2016, pp. 5906–5911.

[29] H. Sandberg and A. M. H. Teixeira, "From control system security indices to attack identifiability," in *Proc. Sci. Secur. Cyber-Phys. Syst. Workshop*, 2016, pp. 1–6.

[30] J.-M. Dion, C. Commault, and J. Van Der Woude, "Generic properties and control of linear structured systems: A survey," *Automatica*, vol. 39, no. 7, pp. 1125–1144, 2003.

[31] J. Milošević, H. Sandberg, and K. H. Johansson, "A security index for actuators based on perfect undetectability: Properties and approximation," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2018, pp. 235–241.

[32] A. Krause and D. Golovin, "Submodular function maximization," Tech. Rep., 2014.

[33] F. Bach *et al.*, "Learning with submodular functions: A convex optimization perspective," *Found. Trends Mach. Learn.*, vol. 6, nos. 2/3, pp. 145–373, 2013.

[34] L. Wolsey, "An analysis of the greedy algorithm for the submodular set covering problem," *Combinatorica*, vol. 2, no. 4, pp. 385–393, 1982.

[35] G. Nemhauser, L. Wolsey, and M. Fisher, "An analysis of approximations for maximizing submodular set functions–I," *Math. Program.*, vol. 14, no. 1, pp. 265–294, 1978.

[36] M. Zeller, "Myth or reality—Does the Aurora vulnerability pose a risk to my generator?" in *Proc. 37th Annu. Western Protective Relay Conf.*, 2011, pp. 130–136.

[37] J. W. Simpson-Porco, F. Dörfler, and F. Bullo, "Synchronization and power sharing for droop-controlled inverters in islanded microgrids," *Automatica*, vol. 49, no. 9, pp. 2603–2611, 2013.

[38] M. Amin and P. F. Schewe, "Preventing blackouts," *Sci. Am.*, vol. 296, no. 5, pp. 60–67, 2007.

[39] O. C. Imer, S. Yuksel, and T. Başar, "Optimal control of LTI systems over unreliable communication links," *Automatica*, vol. 42, no. 9, pp. 1429–1439, 2006.

[40] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Revealing stealthy attacks in control systems," in *Proc. Allerton Conf. Commun., Control, Comput.*, 2012, pp. 1806–1813.

[41] V. Tzoumas, A. Jadbabaie, and G. J. Pappas, "Sensor placement for optimal Kalman filtering: Fundamental limits, submodularity, and algorithms," in *Proc. Am. Control Conf.*, 2016, pp. 191–196.

[42] M. Stoer and F. Wagner, "A simple min-cut algorithm," *J. ACM*, vol. 44, no. 4, pp. 585–591, 1997.

[43] E. Tegling and H. Sandberg, "On the coherence of large-scale networks with distributed PI and PD control," *IEEE Control Syst. Lett.*, vol. 1, no. 1, pp. 170–175, Jul. 2017.

[44] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Syst. Mag.*, vol. 35, no. 1, pp. 93–109, Feb. 2015.

[45] T. Cormen, *Introduction to Algorithms*. Cambridge, MA, USA: MIT Press, 2009.

[46] A. R. Bergen and D. J. Hill, "A structure preserving model for power system stability analysis," *IEEE Trans. Power App.Syst.*, vol. PAS-100, no. 1, pp. 25–35, Jan. 1981.

[47] S. K. M. Kodsi and C. A. Canizares, "Modeling and simulation of IEEE 14-bus system with facts controllers," Tech. Rep., Univ. Waterloo, Waterloo, ON, Canada, 2003.

[48] A. Mohsenian-Rad and A. Leon-Garcia, "Distributed internet-based load altering attacks against smart power grids," *IEEE Trans. Smart Grid*, vol. 2, no. 4, pp. 667–674, Dec. 2011.

[49] H. Trentelman, A. Stoorvogel, and M. Hautus, *Control Theory for Linear Systems*. Berlin, Germany: Springer, 2012.

[50] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM Rev.*, vol. 51, no. 1, pp. 34–81, 2009.

Jezdimir Milošević received the M.Sc. degree in electrical engineering and computer science in 2015 from the School of Electrical Engineering, University of Belgrade, Belgrade, Serbia. He is currently working toward the Ph.D. degree with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden.

He was a Visiting Researcher with the University of Hawaii at Manoa in 2014, and Massachusetts Institute of Technology in 2018 and 2019. His research interests are within cyber-security of industrial control systems.

André Teixeira received the M.Sc. degree in electrical and computer engineering from the Faculty of Engineering, University of Porto, Porto, Portugal, in 2009, and the Ph.D. degree in automatic control from the KTH Royal Institute of Technology, Stockholm, Sweden, in 2014.

He is an Associate Senior Lecturer with the Division of Signals and Systems, Department of Engineering Sciences, Uppsala University, Uppsala, Sweden. From 2014 to 2015, he was a Postdoctoral Researcher with the Department of Automatic Control, KTH Royal Institute of Technology. From October 2015 to August 2017, he was an Assistant Professor with the Faculty of Technology, Policy and Management, Delft University of Technology.

Karl H. Johansson received the M.Sc. and Ph.D. degrees from Lund University, Lund, Sweden, in 1992 and 1997, respectively.

He is the Director of the Stockholm Strategic Research Area ICT The Next Generation and Professor with the School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden. He has held visiting positions with UC Berkeley, Caltech, NTU, HKUST Institute of Advanced Studies, and NTNU. His research interests include networked control systems, cyber-physical systems, and applications in transportation, energy, and automation.

Dr. Johansson is a member of the IEEE Control Systems Society Board of Governors, the IFAC Executive Board, and the European Control Association Council. He has received several best paper awards and other distinctions. He has been awarded Distinguished Professor with the Swedish Research Council and Wallenberg Scholar. He has received the Future Research Leader Award from the Swedish Foundation for Strategic Research and the triennial Young Author Prize from IFAC. He is fellow of the Royal Swedish Academy of Engineering Sciences, and a IEEE Distinguished Lecturer.

Henrik Sandberg received the M.Sc. degree in engineering physics and the Ph.D. degree in automatic control from Lund University, Lund, Sweden, in 1999 and 2004, respectively.

He is a Professor with the Division of Decision and Control Systems, KTH Royal Institute of Technology, Stockholm, Sweden. From 2005 to 2007, he was a Postdoctoral Scholar with the California Institute of Technology, Pasadena, CA, USA. In 2013, he was a Visiting Scholar with the Laboratory for Information and Decision Systems, Massachusetts Institute of Technology, Cambridge, MA, USA. He has also held visiting appointments with the Australian National University, Canberra, ACT, USA, and the University of Melbourne, Parkville, VIC, Australia. His current research interests include security of cyber-physical systems, power systems, model reduction, and fundamental limitations in control.

Dr. Sandberg received the Best Student Paper Award from the IEEE Conference on Decision and Control in 2004, an Ingvar Carlsson Award from the Swedish Foundation for Strategic Research in 2007, and Consolidator Grant from the Swedish Research Council in 2016. He has served on the editorial board of IEEE TRANSACTIONS ON AUTOMATIC CONTROL and is currently Associate Editor of the IFAC Journal *Automatica*.