This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

A Graph Theoretic Characterization of Perfect Attackability for Secure Design of Distributed Control Systems

Sean Weerakkody Xiaofei Liu Sang H. Son Bruno Sinopoli

***Abstract*—This article considers secure design in distributed control systems to ensure the detection of stealthy integrity attacks. Distributed control systems consist of many heterogeneous components such as sensors, controllers, and actuators and may contain several independent agents. The presence of many components and agents in a system increases the attack surfaces for potential adversaries, making distributed control systems vulnerable to malicious behavior. The goal of this article is to consider the design of distributed control systems to ensure the deterministic detection of attacks. To do this, we leverage existing results which relate the deterministic detection of a fixed set of malicious nodes to structural left invertibility. We extend the notion of structural left invertibility to consider attacks from all possible sets of malicious nodes using vertex separators. Vertex separators are then used to solve optimization problems which aim to minimize communication networks while also ensuring that a resource limited adversary cannot generate perfect attacks. Optimal bounds on communication and sensing are obtained and polynomial time design algorithms are provided.**

I. Introduction

Distributed control systems (DCS), systems where controllers are distributed throughout a network, have become ubiquitous in today's society. In a DCS, systems leverage the presence of multiple agents and controllers, possibly connected by a communication network, to manipulate many local environments and meet global objectives. With next generation improvements in networking, sensing, and computing as well as society's reliance on large scale systems, DCS are present in a variety of applications. These include the smart grid [1], [2], sensor networks [3], formation control of autonomous vehicles [4], [5], average consensus [6], and process plants.

However, because DCS often rely on heterogeneous components and off the shelf networking and may potentially depend on the actions of multiple independent agents, they are vulnerable to malicious attacks [7]. Moreover since DCS are linked to society's critical infrastructures, their security is of paramount importance. Serious breaches in control systems have already occurred including Stuxnet, an attack on an uranium plant in Iran, and the Maroochy Shire incident where a disgruntled employee disrupted sewage management [8].

Recent research efforts have aimed to determine when systems are vulnerable to undetectable attacks in order to motivate resilient design. For instance, Liu et al. in [9] provide algebraic conditions linked to the design of the electricity grid under which a knowledgeable adversary can cause errors in state estimation. Sandberg et al. in [10] propose multiple security indices for sensors which allow a system operator to identify sparse attacks on the grid. Additionally, Mo et al. in [11] and [12] characterize the bias an adversary can introduce into the state estimation error in control systems and sensor networks without incurring detection. The resilience of consensus based algorithms are considered in [13] and [14]. In particular, Sundaram et al. in [13] determine graphical conditions under which a set of agents can compute an arbitrary function of their initial states in the presence of malicious nodes. Additionally, Pasqualetti et al. in [14] characterize attack identifiability and detectability using connectivity and left invertibility.

Once the resilience of systems is characterized, it is important to also consider detection and recovery. For instance, Sundaram et al. [15] consider DCS with a subset of malicious nodes. The authors consider the design of an intrusion detection system which can recover true system outputs as well as identify malicious nodes. Fawzi et al. [16] consider the feasibility of robust control and estimation in the presence of attacks and propose a practical decoder to perform detection.

This paper considers the design of DCS to prevent perfect attacks where an adversary is able to change the state without biasing measurements. In the absence of noise, preventing perfect attacks ensures the deterministic detection of adversarial behavior. Cam et al. [17] characterize DCS that are perfectly attackable. Moreover, Pasqualetti et al. [18] use structural left invertibility to graphically determine if a system is vulnerable to perfect attacks from a fixed set of malicious nodes.

In this article, we consider the setting of DCS where no more than p agents and sensors may be compromised. We assume the adversary wishes to carry out a perfect attack so he can disrupt the system without being detected by a system operator who can impose countermeasures. We assume a centralized detector collects measurements from a subset of the agents and is aware of each agent's control policy. Given these assumptions, we extend the results of [17] and [18] by graphically characterizing systems which are not perfectly attackable for all feasible sets of malicious nodes. To do this, our first contribution extends the notion of structural left invertibility, used to described the resilience of DCS to attacks from a fixed set of malicious nodes [18], to the notion of vertex separators which can consider all feasible sets of malicious

S Weerakkody, X. Liu, and B. Sinopoli are with the Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, USA 15213. Email: {sweerakk,xiaofeil}@andrew.cmu.edu, brunos@ece.cmu.edu,

S. Weerakkody is supported in part by the Department of Defense (DoD) through the National Defense Science & Engineering Graduate Fellowship (NDSEG) Program. The work by S. Weerakkody, X. Liu, and B. Sinopoli is supported in part by the Department of Energy under Award Number DE-OE0000779 and by the National Science Foundation under Award Number CNS-1329936

S. Son is with the Department of Information and Communication Engineering, Daegu Gyeongbuk Inst. Of Science and Technology (DGIST), Dalseong-Gun, Daegu, Korea. Email: son@dgist.ac.kr

S. Son is supported in part by the Global Research Laboratory Program (2013K1A1A2A02078326) through NRF, and Institute for Information & communications Technology Promotion (IITP) grant funded by the Korean government (MSIP) (No. B0101-15-0557, Resilient Cyber-Physical Systems Research)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

nodes. This result allows us to determine in polynomial time if a system is robust to perfect attacks.

Our second main contribution is to formulate and solve optimization problems which minimize sensing and communication in DCS while ensuring resilience to perfect attacks. We first consider an unconstrained minimization problem, where there are no restrictions on which agents may communicate or be observed. For a fixed number of observers, we find the minimum number of communication links that can guarantee perfect detectability. Furthermore, we completely characterize the subset of networks which solve the optimization problem and contain no cycles among unobserved agents. We then show the problem of jointly minimizing the number of sensors and communication links strictly depends upon the cost of sensing and communicating. If sensing is more expensive, we show that it is optimal to use the same number of sensors as malicious nodes. If communicating is more expensive, it is optimal to deploy a sensor for every agent.

We next consider a constrained version of this optimization problem, where the set of agents which can communicate and be observed is restricted. If resilient design to prevent perfect attacks is feasible, then we show that the minimum number of communication links is the same as the unconstrained case. Moreover, we demonstrate that obtaining a feasible network configuration is equivalent to solving maximum flow problems for each agent in the system. Here, when jointly minimizing sensing and communication in the constrained case, we show that if sensing is more expensive, the optimal policy is to use the fewest number of sensors that guarantee a system is not susceptible to perfect attacks.

A preliminary version of this article [19] was recently submitted. We extend these results by first obtaining graphical solutions to the unconstrained optimization problem in the special case where there are no cycles among unobserved agents in the DCS. Secondly, we consider the design of DCS under constraints on sensing and the communication network which is a realistic concern in large scale distributed control systems. We offer polynomial time algorithms to generate optimal networks which are not perfectly attackable and thus ensure deterministic detection of malicious behavior.

The remainder of the paper is formulated as follows. In section II, we provide descriptions of our distributed control system. In section III, we introduce an attack model, define perfectly attackable systems and revisit structural left invertibility. In section IV, we provide graphical conditions for a system to be perfectly attackable using vertex separators. In section V, we formulate optimization problems to minimize the amount of communication in the system while ensuring that the network is not perfectly attackable. This ensures deterministic detection of an adversary. In section VI, we formulate and solve this optimization problem in the case of constrained communication and sensing among agents. Section VII contains a simulation and section VIII concludes the paper.

II. System Model

In this section, we introduce the model used to describe DCS. We assume that there are n agents, $x_1, \cdots, x_n$ communicating with each other, and that they are observed by m observers, $y_1, \cdots, y_m$ where $m \leq n$. We let $\mathcal{X} \triangleq \{x_1, \cdots, x_n\}$ and $\mathcal{Y} \triangleq \{y_1, \cdots, y_m\}$ and model interactions between agents and observers using a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ where $\mathcal{V} = \mathcal{X} \cup \mathcal{Y}$ is the set of agents and observers. In addition, $\mathcal{E} \subset \mathcal{V} \times \mathcal{V}$ represents communication between agents or observation of an agent by an observer. That is, if $(x_i, x_j) \in \mathcal{E}$, then x_i may send messages to x_j. Moreover, if $(x_j, y_i) \in \mathcal{E}$, then observer y_i can measure the state of agent x_j. We assume that every agent in $\mathcal{X}$ has a self loop so that $(x_i, x_i) \in \mathcal{E}$ for all $i = 1, \cdots, n$.

A. Graph Theory Preliminaries

The set of incoming neighbors to a node v_i is given as

$$N^I_{v_i} = \{v_j \in \mathcal{V} :\ (v_j, v_i) \in \mathcal{E}\}. \tag{1}$$

The in-degree of node v_i, denoted as $d^I_{v_i}$ is given by $d^I_{v_i} = |N^I_{v_i}|$. Similarly, the set of outgoing neighbors is defined as

$$N^O_{v_i} = \{v_j \in \mathcal{V} :\ (v_i, v_j) \in \mathcal{E}\}. \tag{2}$$

The out-degree $d^0_{v_i}$ of node v_i is given by $d^O_{v_i} = |N^O_{v_i}|$.

Suppose $A \subset \mathcal{V}$ and $B \subset \mathcal{V}$. A path from A to B is a sequence of vertices $v_1, v_2, \cdots, v_l$ where $v_1 \in A$, $v_l \in B$, and $(v_j, v_{j+1}) \in \mathcal{E}$ for $1 \leq j \leq l-1$. A simple path contains no repeated vertices. Two paths are vertex disjoint, if they contain no common vertices. For simplicity, in this article two paths which are vertex disjoint will simply be referred to as disjoint. Two paths are internally disjoint if they have no common vertices except for possibly the starting and ending vertices. In general l paths are disjoint if any pair of paths are disjoint. A set of l disjoint and simple paths from a set A to a set B is called a l-linking from A to B.

B. Control and Sensing

When the system is operating normally, we assume that each agent $x_i \in \mathcal{X}$ is associated with a scalar state $\mathbf{x}_i(k)$ which is dependent on time. The state dynamics for agent x_i is given by

$$\mathbf{x}_i(k+1) = a_{i,i}\mathbf{x}_i(k) + \mathbf{u}_i(k) + \mathbf{w}_i(k). \tag{3}$$

$\mathbf{w}_i(k)$ is the process noise in the system and $\mathbf{u}_i(k)$ is the input to the agent. We assume that $\mathbf{u}_i(k)$ can be written as follows

$$\mathbf{u}_i(k) = \sum_{\substack{j \neq i \\ x_j \in (N^I_{x_i} \cap \mathcal{X})}} a_{i,j}\mathbf{x}_j(k). \tag{4}$$

Thus, agent x_i receives the state from each of his incoming neighbors at each time step and computes a control law, which is a linear function of his incoming neighbor's states.

Remark 1. *Although we assume that each input is a linear function of the state, in general we can consider an input of the form $\mathbf{u}_i(k) = \sum_{j \neq i, x_j \in (N^I_{x_i} \cap \mathcal{X})} a_{i,j}\mathbf{x}_j(k) + \mathbf{u}^*_i(k)$, as long as the anomaly detection center, discussed later, knows $\mathbf{u}^*_i(k)$ for all $x_i \in \mathcal{X}$ and for all k. Additionally, the state $\mathbf{x}_i(k)$ can refer to a physical quantity such as velocity, or can simply be a number associated with the agent, for instance a value used for consensus.*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

As mentioned earlier, a set of observers $\mathcal{Y}$ is used to measure the state of a portion of the agents. We assume each observer measures exactly one agent, and no two observers measure the same agent. If observer y_i measures agent x_j,

$$\mathbf{y}_i(k) = \mathbf{x}_j(k), \tag{5}$$

where $\mathbf{y}_i(k)$ is the value of the measurement at time k from observer y_i.

Remark 2. *The assumption of dedicated sensors is based on the fact that the system is distributed and it's likely that sensors would not have the physical access to measure multiple agents accurately. While we assume there are no redundant sensors, in practice multiple sensors can be added to measure a single state in the physical system for robustness. However, for our treatment, it is likely that if an attacker can manipulate one sensor that measures a state x_j, he has the ability to access and manipulate all sensors that measure the state x_j, especially if the hardware itself is redundant. As a result, for the purposes of modeling attacks, we can consider redundant sensors as a single node.*

Remark 3. *We assume there is no sensor noise in* (5). *The existence of sensor noise prevents an agent from sending its exact state in* (4). *We note that modeling sensor noise does not affect subsequent analysis in this article which assume the existence of perfect attacks is related to system left invertibility. Thus, for simplicity, sensor noise is not considered.*

To simplify notation, we define vectors

$$\begin{aligned}
\mathbf{x}(k) &\triangleq \begin{bmatrix}\mathbf{x}_1(k) & \cdots & \mathbf{x}_n(k)\end{bmatrix}^T \in \mathbb{R}^n,\\
\mathbf{w}(k) &\triangleq \begin{bmatrix}\mathbf{w}_1(k) & \cdots & \mathbf{w}_n(k)\end{bmatrix}^T \in \mathbb{R}^n,\\
\mathbf{y}(k) &\triangleq \begin{bmatrix}\mathbf{y}_1(k) & \cdots & \mathbf{y}_m(k)\end{bmatrix}^T \in \mathbb{R}^m.
\end{aligned}$$

The dynamics in the full DCS is given by

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + \mathbf{w}(k), \quad \mathbf{y}(k) = C\mathbf{x}(k), \tag{6}$$

where $A \triangleq [a_{i,j}]$ and the matrix C is defined entrywise as

$$C_{ij} = \mathbf{1}_{(x_j, y_i) \in \mathcal{E}}, \tag{7}$$

where $\mathbf{1}$ is the indicator function. To ensure that the state remains bounded, the matrix A is designed to be at least marginally stable so $\rho(A) \leq 1$ and any eigenvalue with magnitude 1 has Jordan block of size 1.

C. Centralized Detection

A centralized anomaly detector is used to detect irregularities in the system. The anomaly detector receives sensor measurements and uses a linear filter to perform estimation. We assume the detector knows A and C so that it is aware of the structure of the DCS as well as each agent's control law. Furthermore, if an agent changes its update rule, it notifies the detector. The centralized detector uses the linear filter

$$\hat{\mathbf{x}}(k+1) = (A - KCA)\hat{\mathbf{x}}(k) + K\mathbf{y}(k+1), \tag{8}$$

to estimate the state, where $(A - KCA)$ is stable. Here, $\hat{\mathbf{x}}(k)$ is an estimate of the state $\mathbf{x}(k)$. A χ^2 detector is used to detect abnormalities or attacks. In particular, an alarm is triggered if

$$\mathbf{z}(k)^T \mathcal{P}^{-1} \mathbf{z}(k) > \eta, \tag{9}$$

where $\mathbf{z}(k) = \mathbf{y}(k) - CA\hat{\mathbf{x}}(k-1)$ is the residue, $\mathcal{P}$ is the covariance of the residue, and η is the threshold. Thus, if measurements significantly deviate from their expected values, an alarm is raised.

III. Attack Model

A. Graphical Attack Description

In this section, we consider the model of a DCS under attack. We assume that at time 0 a subset of nodes in $\mathcal{V}$ is compromised. The set of compromised nodes may consist of both agents and observers. Considering both sensor and agent attacks is reasonable since if an adversary is able to corrupt an agent, it may also have access to the sensor which measures the agent. For instance, in [20], it was shown that an adversary who hacks a vehicle's CAN bus can manipulate both steering as well as speedometer readings.

The set of compromised nodes are denoted by $F \subset \mathcal{V}$. While the set of compromised nodes is unknown to the anomaly detector, it is known that $|F| \leq p$. This corresponds to the defender considering how many malicious nodes it is willing to tolerate. The set of all feasible sets of compromised nodes is given by

$$\mathcal{F} = \{F \subset \mathcal{V}, |F| \leq p\}. \tag{10}$$

For a fixed set of compromised nodes $F = \{x_{i_1}, \cdots, x_{i_l}, y_{i_{l+1}}, \cdots y_{i_{p'}}\}$ with $|F| = p' \leq p$, we introduce a corresponding set of attack nodes in our graph $\mathcal{U}_F \triangleq \{u_1^a, \cdots u_{p'}^a\}$, where u_j^a is a dedicated attack input of x_{i_j} for $j \leq l$ and a dedicated attack input of y_{i_j} for $j > l$.

We model the system under attack by nodes in F with a directed graph $\mathcal{G}_F^a = (\mathcal{V}_F^a, \mathcal{E}_F^a)$. In this case, $\mathcal{V}_F^a = \mathcal{V} \cup \mathcal{U}_F$ and $\mathcal{E}_F^a = \mathcal{E} \cup \mathcal{E}_{\mathcal{U}_F, \mathcal{X}} \cup \mathcal{E}_{\mathcal{U}_F, \mathcal{Y}}$ where

$$\mathcal{E}_{\mathcal{U}_F, \mathcal{X}} = \{(u_1^a, x_{i_1}), \cdots, (u_l^a, x_{i_l})\}, \tag{11}$$

$$\mathcal{E}_{\mathcal{U}_F, \mathcal{Y}} = \{(u_{l+1}^a, y_{i_{l+1}}), \cdots (u_{p'}^a, y_{i_{p'}})\}. \tag{12}$$

As a result, we now add additional attack nodes to our system digraph to represent malicious inputs to compromised sensors or agents.

B. Algebraic Attack Description

We let $\mathbf{x}^a(k)$ denote the state of the system under attack and $\mathbf{y}^a(k)$ denote the corresponding output. We assume the adversary injects additive attacks. Thus, if agent x_i is compromised by an input u_l^a, we assume its update rule follows

$$\mathbf{x}_i^a(k+1) = a_{i,i}\mathbf{x}_i^a(k) + \sum_{\substack{j \neq i \\ x_j \in (N_{x_i}^I \cap \mathcal{X})}} a_{i,j}\mathbf{x}_j^a(k) + \mathbf{u}_l^a(k) + \mathbf{w}_i(k). \tag{13}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

where $\mathbf{u}_l^a(k)$ is the attack input chosen by u_l^a at time k. If an agent x_i is not compromised, the update rule is

$$\mathbf{x}_i^a(k+1) = a_{i,i}\mathbf{x}_i^a(k) + \sum_{\substack{j \neq i \\ x_j \in (N_{x_i}^I \cap \mathcal{X})}} a_{i,j}\mathbf{x}_j^a(k) + \mathbf{w}_i(k). \quad (14)$$

If an observer y_i measuring $\mathbf{x}_j$ is compromised by input u_l^a, then its measurement is given by

$$\mathbf{y}_i^a(k) = \mathbf{x}_j^a(k) + \mathbf{u}_l^a(k). \quad (15)$$

An uncompromised sensor measures the true state of an agent as follows

$$\mathbf{y}_i^a(k) = \mathbf{x}_j^a(k). \quad (16)$$

Finally, to simplify notation, we define $B_F^a \in \mathbb{R}^{n \times p'}$ and $D_F^a \in \mathbb{R}^{m \times p'}$ entrywise as

$$B_F^a(i,j) = \mathbf{1}_{(u_j^a, x_i) \in \mathcal{E}_{\mathcal{U}_F, \mathcal{X}}}, \quad D_F^a(i,j) = \mathbf{1}_{(u_j^a, y_i) \in \mathcal{E}_{\mathcal{U}_F, \mathcal{Y}}}. \quad (17)$$

Also, let $\mathbf{u}^a(k) = \begin{bmatrix} \mathbf{u}_1^a(k) & \cdots & \mathbf{u}_{p'}^a(k) \end{bmatrix}^T$. Thus, when under attack, the state dynamics of the DCS is given by

$$\mathbf{x}^a(k+1) = A\mathbf{x}^a(k) + B_F^a\mathbf{u}^a(k) + \mathbf{w}(k), \quad (18)$$

$$\mathbf{y}^a(k) = C\mathbf{x}^a(k) + D_F^a\mathbf{u}^a(k). \quad (19)$$

The estimator policy is unchanged as shown below

$$\hat{\mathbf{x}}^a(k+1) = (A - KCA)\hat{\mathbf{x}}^a(k) + K\mathbf{y}^a(k+1), \quad (20)$$

$$\mathbf{z}^a(k) = \mathbf{y}^a(k) - CA\hat{\mathbf{x}}^a(k-1). \quad (21)$$

C. Attack Strategy and Perfect Attacks

We assume that the goal of an adversary in a DCS is to affect the state of the distributed system without raising an alarm in the anomaly detection center. Such a strategy allows the adversary to implement his attack for long periods of time without action from the defender to prevent damage to the DCS.

To begin, consider the difference between the compromised system and the system operating normally. We define the following variables which measure the difference between the normal and attacked system.

$$\Delta\mathbf{x}(k) \triangleq \mathbf{x}(k) - \mathbf{x}^a(k), \quad \Delta\hat{\mathbf{x}}(k) \triangleq \hat{\mathbf{x}}(k) - \hat{\mathbf{x}}^a(k), \quad (22)$$

$$\Delta\mathbf{y}(k) \triangleq \mathbf{y}(k) - \mathbf{y}^a(k), \quad \Delta\mathbf{z}(k) \triangleq \mathbf{z}(k) - \mathbf{z}^a(k). \quad (23)$$

The goal of the adversary is to introduce a large change, $\Delta\mathbf{x}(k)$, in the state without raising an alarm. From the choice of a χ^2 detector, the adversary wants to minimize the magnitude of $\Delta\mathbf{z}(k)$. Ideally, the attacker would have no impact on the residue, in which case the attack is perfect as described below.

Definition 4. *An attack is perfect if $\Delta\mathbf{z}(k) = 0$ for all $k \geq 0$ and $\mathbf{u}^a(k) \neq 0$ for some $k \geq 0$. A system (A, C) is perfectly attackable if there exists a set of compromised nodes $F \in \mathcal{F}$ which ensure the existence of a perfect attack.*

Remark 5. *If an attack is perfect, both the measurement and residues of the DCS are unaffected by the adversary's actions. As a result, the adversary is able to bias the state away from the region of desired operation without incurring detection. For deterministic control systems with known initial state, designing a system which is not perfectly attackable is both necessary and sufficient for the deterministic detection of malicious behavior. However, in practice $\Delta\mathbf{z}(k)$ need not be 0 to avoid detection. Cam et al. [17], for instance, consider the notion of a nearly perfect attack where an adversary can destabilize a system with bounded effect on the residues, while Mo et al. [11] and [12] examine the impact of such an attacker. In this paper, however, we restrict our attention to perfect attacks, based on the observation that preventing perfect attacks is a necessary condition for secure DCS design.*

We now briefly review both algebraic and graphical conditions which allow for a system to be perfectly attackable. To begin we introduce left invertibility.

Definition 6. *We define a system (A, B_F^a, C, D_F^a) with a fixed set of attack vertices F to be left invertible if for the following system*

$$\mathbf{x}(k+1) = A\mathbf{x}(k) + B_F^a\mathbf{u}^a(k), \quad \mathbf{y}(k) = C\mathbf{x}(k) + D_F^a\mathbf{u}^a(k), \quad (24)$$

with initial condition $\mathbf{x}(0) = 0$, $\mathbf{y}(k) = 0$ for all k implies that $\mathbf{u}^a(k) = 0$ for all k.

It can be shown that the left invertibility is a necessary and sufficient condition for a system (A, B_F^a, C, D_F^a) with compromised nodes F to avoid perfect attacks. In particular, we have the following theorem from [17], trivially extended to consider both sensor and agent attacks.

Theorem 7. *The following statements are equivalent.*

1) *There exists a sequence of inputs $\mathbf{u}^a(k) \neq 0$ such that $\Delta\mathbf{z}(k) = 0$ for all k.*
2) *There exists a sequence of inputs $\mathbf{u}^a(k) \neq 0$ such that $\Delta\mathbf{y}(k) = 0$ for all k.*
3) *(A, B_F^a, C, D_F^a) is not left invertible.*
4) *The transfer function $C(zI - A)^{-1}B_F^a + D_F^a$ has normal rank less than p'.*

Remark 8. *We can perform perfect state estimation in a deterministic system if the system is left invertible and the initial state x_0 and set of malicious nodes are known. However, identifying the set of malicious nodes is a harder problem than the ensuring deterministic detectability [18]. Previous work, however, has considered the problem of robust estimation, for instance see [16], [21], [22].*

We now use structural systems [23] to obtain a graphical characterization. We associate the graph $\mathcal{G}_F^a$ with a tuple of structural matrices $([A], [B_F^a], [C], [D_F^a])$. We observe that $\mathcal{E}_F^a = \mathcal{E}_{\mathcal{X},\mathcal{X}} \cup \mathcal{E}_{\mathcal{U}_F,\mathcal{X}} \cup \mathcal{E}_{\mathcal{X},\mathcal{Y}} \cup \mathcal{E}_{\mathcal{U}_F,\mathcal{Y}}$ where $\mathcal{E}_{\mathcal{X},\mathcal{X}} = \{(x_i, x_j) : [A]_{ji} \neq 0\}$, $\mathcal{E}_{\mathcal{U}_F,\mathcal{X}} = \{(u_i, x_j) : [B_F^a]_{ji} \neq 0\}$, $\mathcal{E}_{\mathcal{X},\mathcal{Y}} = \{(x_i, y_j) : [C]_{ji} \neq 0\}$, and $\mathcal{E}_{\mathcal{U}_F,\mathcal{Y}} = \{(u_i, y_j) : [D_F^a]_{ji} \neq 0\}$. Also $[A]_{ij} \neq 0$ means that A_{ij} is a free parameter while $[A]_{ij} = 0$ implies that A_{ij} is fixed to be 0.

As a result, the graph $\mathcal{G}_F^a$ obtained under attack for a fixed set of vulnerable nodes F is precisely the graph associated with the structural system $([A], [B_F^a], [C], [D_F^a])$. Also observe that $\mathcal{G} = (\mathcal{V}, \mathcal{E}_{\mathcal{X},\mathcal{Y}} \cup \mathcal{E}_{\mathcal{X},\mathcal{X}})$ so that $\mathcal{G}$ is associated with the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

structural system $([A],[C])$. We now use structural systems to obtain a graphical characterization which is almost always equivalent to left invertibility. In particular, we have the following definition

Definition 9. *The structural system $([A],[B_F^a],[C],[D_F^a])$ is structurally left invertible if there exists an admissible realization of (A,B_F^a,C,D_F^a) that is left invertible.*

We note that if $([A],[B_F^a],[C],[D_F^a])$ is structurally left invertible, then every realization is left invertible with the exception of a set of measure 0. Moreover, if $([A],[B_F^a],[C],[D_F^a])$ is not structurally left invertible, then every admissible realization of (A,B_F^a,C,D_F^a) is also not left invertible. Recall a set of l disjoint and simple paths from a set A to a set B is called a l-linking from A to B. We have the following result that characterizes structural left invertibility from [18].

Theorem 10. *The system $([A],[B_F^a],[C],[D_F^a])$ is structurally left invertible if and only if there exists a linking of size $|\mathcal{U}_F| = p'$ from $\mathcal{U}_F$ to $\mathcal{Y}$.*

From this theorem we see that in order to design a system that ensures deterministic detection of an adversary, more sensors are required than potential attack inputs. Consequently, we have the following result.

Corollary 11. *The system $([A],[B_F^a],[C],[D_F^a])$ is structurally left invertible only if $m \geq p'$.*

IV. Vertex Separators and Structural Left Invertibility

In the previous section we showed that for almost all realizations of a structural system $([A],[B_F^a],[C],[D_F^a])$ with graph $\mathcal{G}_F^a$, structural left invertibility is equivalent to the nonexistence of perfect attacks from malicious nodes in F. In general, we would like to obtain a necessary and sufficient condition which ensures a system is not perfectly attackable. This implies deterministic detection of adversarial behavior for all possible sets of malicious nodes $F \in \mathcal{F}$. Thus, given $([A],[C])$ and corresponding graphical realization $\mathcal{G}$, we would like to determine if there exists a perfect attack for some feasible set of vulnerable nodes $F \in \mathcal{F}$.

In this section, we use vertex separators to characterize DCS with graphical realization given by $([A],[C])$ that are structurally left invertible for all feasible attacks and thus guarantee deterministic detection of adversaries. We begin by defining vertex separators.

Definition 12. *Given a graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$, a vertex separator $S \subset \mathcal{V}\backslash(a,b)$ of nonadjacent vertices (a,b) is a subset of vertices whose removal deletes all paths from a to b.*

We now consider how vertex separators can be used to characterize systems which are not perfectly attackable. To do this we first define the graph $f(\mathcal{G}) = (\mathcal{V} \cup o, \mathcal{E}')$ by adding an additional node o to $\mathcal{G}$ with directed edges from $\mathcal{Y}$ to o. The result follows below.

Theorem 13. *Consider system $([A],[C])$ and corresponding graph realization $\mathcal{G}$ where $m \geq p$ dedicated sensors are assigned to measure a portion of the state. Then $([A],[C])$ is structurally left invertible for all feasible sets of malicious nodes $F \in \mathcal{F}$ if and only if for each agent $x_i \in \mathcal{X}$, all vertex separators S_i of (x_i, o) in $f(\mathcal{G})$ satisfy $|S_i| \geq p$.*

Proof. The proof leverages the following result which relates the notion of a linking used in structural left invertibility to vertex separators. For full details, see [19].

Lemma 14. *Menger's Theorem [24]: the minimum size of a vertex separator between x_i and o is equal to the maximum number of internally disjoint paths from x_i to o.*

Remark 15. *We note that Pasqualetti et al. [14] arrive at necessary conditions for a system to be structurally left invertible using graph connectivity. However, they only consider connected graphs. Using vertex separators, we arrive at a both sufficient and necessary condition for structural left invertibility for all feasible attacks. This resutt illustrates that the digraph need not be connected to ensure that the system is not perfectly attackable.*

We show that determining if a system is perfectly attackable can be done by computing the size of minimal vertex separators between each agent x_i and o. A minimal vertex separator S_i^* of (x_i, o) is a vertex separator containing the fewest number of nodes. It can be shown that computing the size of a minimum vertex separator is equivalent to solving a $0-1$ maximum flow problem [25] on an enlarged graph $h(f(\mathcal{G})) = (\mathcal{V}_H, \mathcal{E}_H)$ where $|\mathcal{V}_H| = 2|\mathcal{V}|$ and $|\mathcal{E}_H| = |\mathcal{E}'| + |\mathcal{V}| - 1$. The graph is obtained by converting every vertex $x \in \mathcal{V} \setminus \{x_i\}$ of $f(\mathcal{G})$ to a directed edge. The Ford-Fulkerson algorithm allows us to compute the size of a minimum vertex separator in $O(|S_i^*||\mathcal{E}_H|)$ time. Moreover, Dinic's algorithm [26], [27], can obtain the size of a minimal vertex separator in $O(|\mathcal{V}_H|^{\frac{1}{2}}|\mathcal{E}_H|)$ time. In order to verify that a system is robust to perfect attacks we must ensure the size of the minimum vertex separator for each $x_i \in \mathcal{X}$ is at least p. Thus, the worse case complexity of system verification using Dinic's algorithm is $O(n|\mathcal{V}_H|^{\frac{1}{2}}|\mathcal{E}_H|)$. This algorithm can also be parallelized over each agent.

The proposed scheme of system verification is a significant improvement compared to existing combinatorial methods. For instance, verifying that every set of feasible attack nodes generates a structurally left invertible graph requires solving $\binom{n+m}{p}$ instances of the maximum flow problem on a graph $h_F(f(\mathcal{G}))$ larger than $h(f(\mathcal{G}))$. Using Dinic's algorithm, this has complexity $O(\binom{n+m}{p}(|\mathcal{V}_H|+2)^{\frac{1}{2}}(|\mathcal{E}_H|+p+1))$. Alternatively, we can compute the rank of matrices as in Theorem 7. However, to enumerate all feasible attacks we must compute the rank of $\binom{n+m}{p}$ matrices in $\mathbb{R}^{m \times p}$.

V. Minimal Design of Systems to Ensure Deterministic Attack Detection

In the previous section we arrived at graphical conditions in terms of vertex separators to determine when a DCS, defined structurally by $([A],[C])$, is perfectly attackable. We now wish to design robust DCS which are not perfectly attackable. In the absence of noise, this ensures deterministic detection of an adversary. However, due to costs of sensing and

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

communication, we also wish to minimize the number of links between agents as well as the number of sensors deployed. Before we introduce our optimization problem, we include the following result.

Lemma 16. *$([A],[C])$ is structurally left invertible for all possible attack configurations only if the out-degree of each node $x_k \in \mathcal{X}$ in $\mathcal{G}$ satisfies $d^O_{x_k} \geq p+1$.*

Proof. From Theorem 13, we only need to find a vertex separator S_i of (x_i, o) in $f(\mathcal{G})$ such that $|S_i| < p$ for some $x_i \in \mathcal{X}$. We choose x_i such that $d^O_{x_i} < p+1$. We argue that a vertex separator S_i of (x_i, o) is the set of x_i's outgoing neighbors not including itself so $S_i = N^O_{x_i} \setminus x_i$. Thus, $|S_i| \leq p-1$. If we remove all the outgoing neighbors of x_i, there is no path from x_i to o and the result holds. □

A. Optimization of Communication

Again consider the graph $f(\mathcal{G}) = (\mathcal{V} \cup o, \mathcal{E}')$ obtained by adding an additional node o to $\mathcal{G}$ with directed edges from $\mathcal{Y}$ to o. Let S_i denote a minimal vertex separator between x_i and o and $\|\cdot\|_0$ denote the number of nonzero entries of a matrix. For fixed m we solve the following problem.

$$\begin{aligned} &\min_{[A],[C]} \|A\|_0 \\ &|S_i| \geq p,\ [A]_{ii} \neq 0,\ \|C^i\|_0 \leq 1 \quad i = 1, \cdots, n \\ &C \in \mathbb{R}^{m \times n},\ \|C_j\|_0 = 1, \quad j = 1, \cdots, m. \end{aligned} \tag{25}$$

C_j denotes the jth row of matrix C and C^i denotes the ith column of C . To reduce the amount of communication, we aim to minimize the number of connections in the system. However, to preserve robustness, we ensure that the system is structurally left invertible for all feasible attacks. We will later show that a system with self loops which is structurally left invertible for all feasible attack policies is also structurally observable. Thus, robust design to guarantee deterministic detection of adversaries simultaneously ensures that the anomaly detection center can obtain an accurate state estimate.

Theorem 17. *The optimal solution to problem 25 is $\|A\|_0^* = mp + (n-m)(p+1) = np + n - m$.*

Proof. We begin by showing that $np + n - m$ is a lower bound of the optimal solution $\|A\|_0^*$. Without loss of generality, assume that $\{x_1, \cdots, x_m\}$ are the set of agents which are observed by $\mathcal{Y}$. Then,

$$\begin{aligned} \|A\|_0^* &= \sum_{k=1}^{m} (d^O_{x_k} - 1) + \sum_{k=m+1}^{n} d^O_{x_k}, \\ &\geq mp + (n-m)(p+1). \end{aligned} \tag{26}$$

The first equality is obtained by noting that the number of nonzero entries in each row i of A is equal to $d^O_{x_k}$ if the agent x_i is unobserved and equal to $d^O_{x_k} - 1$ if it is observed. The last inequality is obtained from the necessary conditions for structural left invertibility described in Lemma 16. Thus $np + n - m$ is a lower bound for $\|A\|_0^*$.

We now show that $np+n-m$ is an upper bound for $\|A\|_0^*$ by constructing a feasible $([A],[C])$ with a minimal number of edges. To do this we consider the following lemma.

Algorithm 1 Find Path from x_i to o when S_i is removed

```
 1: function FIND PATH($\mathcal{G}, x_i$)
 2:     $z = x_i, \quad P = x_i$.
 3:     if $(z, y_j) \in \mathcal{E}$ for some $y_j \in \mathcal{V} \setminus S_i$ then
 4:         $P = P, y_j, o$. return $P$
 5:         break
 6:     end if
 7:     if $\exists\ x_j, y_k \in \mathcal{V} \setminus S_i$ such that $(z, x_j), (x_j, y_k) \in \mathcal{E}$ then
 8:         $P = P, x_j, y_k, o$. return $P$
 9:         break
10:     end if
11:     Find $x_l \in \mathcal{V} \setminus S_i$ such that $(z, x_l) \in \mathcal{E}$ and $(x_l, y_k) \notin \mathcal{E},\ \forall y_k \in \mathcal{Y}$.
12:     $z = x_l, \quad P = P, x_l$.
13:     Proceed to step 7.
14: end function
```

Lemma 18. *Consider a realization $([A],[C])$ of a DCS with graph $\mathcal{G}$ where every nontrivial cycle (cycles not containing self-loops) contains an observed agent. Then $([A],[C])$ is an optimal solution to problem 25 if and only if*

1) *Each agent x_i has out-degree $d^O_{x_i} = p+1$ with $(x_i, x_i) \in \mathcal{E},\ i = 1, \cdots, n$.*
2) *Each observer y_j has out-degree $d^O_{y_j} = 0,\ j = 1, \cdots, m$.*
3) *Each observer y_j has in-degree $d^O_{y_j} = 1,\ j = 1, \cdots, m$ with incoming edge from unique state x_{l_j}.*

We first show there exists a graph $\mathcal{G}$ which satisfies these assumptions. A feasible configuration would be to select m arbitrary agents to observe. WLOG we assume that agents $\{x_1, \cdots, x_m\}$ are observed so that there exists a directed edge from x_j to y_j for $j \in \{1, \cdots, m\}$. Next for $j \in \{1, \cdots, m\}$, we have $d^O_{x_j} = p+1$ and $N^O_{x_j} \subset \{y_j, x_1, \cdots, x_m\}$. Thus, each observed agent has $p+1$ outgoing edges, 1 to its observer, $p-1$ edges to other observed agents, and 1 to itself. Finally, for $j \in \{m+1, \cdots, n\}$, we have $d^O_{x_j} = p+1$ and $N^O_{x_j} \subset \{x_1, \cdots, x_m, x_j\}$. Each unobserved agent has p neighbors besides itself, all of which are observed. Thus, there are no cycles which only contain unobserved agents.

We next show $\mathcal{G}$ satisfies the constraints of problem 25. By inspection, the graph immediately satisfies $[A]_{ii} \neq 0,\ \|C^i\|_0 \leq 1$ for each $i = 1, \cdots, n$ and $C \in \mathbb{R}^{m \times n},\ \|C_j\|_0 = 1$ for each $j = 1, \cdots, m$. We must verify that $S_i \geq p$. Suppose there exists a vertex separator S_i of (x_i, o) in $f(\mathcal{G})$ where $|S_i| < p$. Suppose we remove all vertices S_i from $f(\mathcal{G})$. We can construct a path from x_i to o even when vertices from S_i are removed using Algorithm 1.

Since $|S_i| < p$ and x_i has p outgoing neighbors besides itself, x_i must either be observed by a node $y_j \notin S_i$, have outgoing edge to an observed agent $x_j \notin S_i$ with observer $y_k \notin S_i$, or have outgoing edge to unobserved agent $x_l \notin S_i$. The algorithm terminates successfully if either of the first two conditions hold. Otherwise, the path is extended to the unobserved agent x_l. Since x_l is unobserved, the algorithm proceeds to step 7. The same argument holds for x_l. This

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

process will eventually encounter an observed agent $x_j \notin S_i$ with observer $y_k \notin S_i$ in step 7 since $\mathcal{G}$ is finite and every cycle must contain an observed agent. Consequently, this process will eventually terminate and give a path P from x_i to o. Thus, S_i is not a vertex separator and $\mathcal{G}$ is feasible.

We now show $\mathcal{G}$ constructed with the rules presented in Lemma 18 is optimal. We note that each agent has out-degree $p+1$. Thus, from (26) we have $\|A\|_0 = np + n - m$. Since $np + n - m$ is a lower bound for the number of edges in an optimal solution, $\mathcal{G}$ is an optimal solution. □

The above theorem determines the minimum number of communication links required in a system with m observers to avoid perfect attacks and ensure deterministic detection. While the general graphical solution is unknown, Lemma 18 gives us the structure for optimal graphs that have no nontrivial cycles among unobserved agents.

B. Joint Optimization of Sensing and Communication

Instead of fixing the number of sensors m under consideration, the number of sensors can be a design variable which is chosen concurrently with the network. The adjusted optimization problem is given as

$$\begin{aligned}
&\min_{[A],[C],m} \alpha_1 \|A\|_0 + \alpha_2 m \\
&|S_i| \geq p, \ [A]_{ii} \neq 0, \ \|C^i\|_0 \leq 1, \quad i = 1, \cdots, n \\
&C \in \mathbb{R}^{m \times n}, \ \|C_j\|_0 = 1, \quad j = 1, \cdots, m \\
&m \in \{p, p+1, \cdots, n\}. \qquad (27)
\end{aligned}$$

α_1 is the cost of a communication link and α_2 is the cost of a sensor. This problem is equivalent to

$$\begin{aligned}
&\min_{p \leq m \leq n} \left(\min_{[A],[C]} \alpha_1 \|A\|_0(m) + \alpha_2 m \right), \\
&= \min_{p \leq m \leq n} \alpha_1 n(p+1) + (\alpha_2 - \alpha_1) m.
\end{aligned}$$

Thus if $\alpha_2 > \alpha_1$ so that sensors are more costly than network links we simply take the minimum number of sensors, $m^* = p$. If $\alpha_1 < \alpha_2$, so that network links are more expensive, we take the maximum number of sensors, $m^* = n$. In this case, it is in fact optimal to observe every agent. In a graphical realization of the network, an agent x_i should speak to an arbitrary $p-1$ other agents $x_{j_1}, \cdots, x_{j_{p-1}}$.

VI. Constrained Design of Systems to Ensure Deterministic Attack Detection

A. Constrained Optimization of Communication

In the previous section, we found minimal designs of systems which prevent all possible perfect attacks. In these problems, we assumed that there were no restrictions among which agents can communicate. In practice, due to physical constraints, certain agents may not be able to communicate. We assume constraints on communication are encoded into $[\bar{A}]$ where agent x_i can speak to agent x_j if and only if $[\bar{A}]_{ji} \neq 0$. Given a set of observers $[C]$, we formulate a problem to robustly minimize the amount of communication among agents subject to constraints given by $[\bar{A}]$.

$$\begin{aligned}
&\min_{[A]} \|A\|_0 \\
&|S_i| \geq p, \ [A]_{ii} \neq 0, \quad i = 1, \cdots, n \\
&[\bar{A}]_{uv} = 0 \implies [A]_{uv} = 0. \qquad (28)
\end{aligned}$$

Algorithm 2 Constrained Optimization of DCS

```
1: function OPTIMIZATION(($[\bar{A}], [C]$))
2:     Let graph $\mathcal{G}$ be generated from $[\bar{A}], [C]$, $[A] = [\bar{A}]$.
3:     while $\|A\|_0 > np + n - m$ do
4:         Find an edge $(x_i, x_{i'})$ whose removal still ensures there are no perfect attacks on $\mathcal{G}$.
5:         $\mathcal{G} = \mathcal{G} - (x_i, x_{i'})$, $[A]_{i'i} = 0$.
6:     end while
7: return $[A]$
8: end function
```

We now obtain the following result related to problem 28 which states that if the problem is feasible, there always exists a solution to problem 28 which is also a solution to the unconstrained optimization problem 25.

Theorem 19. *Suppose there exists a feasible solution to problem 28. Then, the optimal solution to problem 28 satisfies $\|A\|_0^* = np + n - m$.*

Proof. We argue that Algorithm 2 can be used to obtain an optimal solution to problem 28. It suffices to show step 4 is feasible for an arbitrary $\mathcal{G}$ which is not perfectly attackable and is non-minimal. To do this, we observe there must exist an agent x_i with out-degree $d^O_{x_i} > p + 1$ if the system is non-minimal. Since the system is not perfectly attackable, by Lemma 14, there exists at least p disjoint paths from x_i to o. Because x_i has out-degree greater than $p+1$, there exists an edge $(x_i, x_{i'})$ whose removal ensures x_i still has p disjoint paths to o so that $|S_i| \geq p$ in $f(\mathcal{G} - (x_i, x_{i'}))$.

Now consider arbitrary x_j not equal to x_i in $\mathcal{G}$. We must show that $|S_j| \geq p$ where S_j is a minimum vertex separator of x_j and o in $f(\mathcal{G} - (x_i, x_{i'}))$. Suppose $|S_j| < p$. We observe that $\{S_j, x_i\}$ is a vertex separator of x_j and o in $f(\mathcal{G})$. Since $\mathcal{G}$ is not perfectly attackable, $|\{S_j, x_i\}| \geq p$. Consequently, $x_i \notin S_j$, $|S_j| = p - 1$, and $\{S_j, x_i\}$ is a minimal vertex separator of (x_j, o) in $f(\mathcal{G})$.

Lets remove S_j from $f(\mathcal{G} - (x_i, x_{i'}))$. We first argue there must still be a path from x_j to x_i. Suppose instead that removing S_j from $f(\mathcal{G} - (x_i, x_{i'}))$ deletes all paths from x_j to x_i. Then, removing S_j from $f(\mathcal{G})$ deletes all paths from x_j to x_i in $f(\mathcal{G})$. Since $\{S_j, x_i\}$ is a minimal vertex separator of (x_j, o) in $f(\mathcal{G})$, removing S_j from $f(\mathcal{G})$ means there is still a path from x_j to o. In addition, removing x_i does not delete this path to o since there is no path from x_j to x_i after deleting S_j. This contradicts $\{S_j, x_i\}$ as a vertex separator of x_j and o in $f(\mathcal{G})$. By contradiction, there must still be a path from x_j to x_i after deleting S_j from $f(\mathcal{G} - (x_i, x_{i'}))$.

We now show there exists a path from x_i to o after removing S_j from $f(\mathcal{G} - (x_i, x_{i'}))$. By assumption, there are at least p disjoint paths from x_i to o in $f(\mathcal{G} - (x_i, x_{i'}))$. Deleting S_j,

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

Algorithm 3 Practical Solution to Constrained Optimization of DCS

```
1:  function OPTIMIZATION(($[\bar{A}], [C]$))
2:      Let graph $\mathcal{G}$ be generated from $[\bar{A}], [C]$, $[A] = [\bar{A}]$.
3:      for $i = 1 : n$ do
4:          if $d^O_{x_i} > p + 1$ then
5:              Use Dinic's [26], [27] (or other max-flow) algorithm to obtain a maximum linking from $x_i$ to $o$
6:              Keep $p$ neighbors through which $\exists$ $p$ vertex disjoint paths from $x_i$ to $o$, not containing $x_i$.
7:              Update $\mathcal{G}$ and $[A]$.
8:          end if
9:      end for
10: return $[A]$
11: end function
```

which has $p - 1$ vertices, can remove at most $p - 1$ paths. Thus, there is still a path from x_i to o.

As a result, even after deleting S_j from $f(\mathcal{G} - (x_i, x_{i'}))$, there exists a path from x_j to x_i and a path from x_i to o. Consequently, there exists a path from x_j to o so that S_j is not a vertex separator. Thus, by contradiction, any vertex separator S_j of (x_j, o) in $f(\mathcal{G} - (x_i, x_{i'}))$ satisfies $|S_j| \geq p$. Therefore, $\mathcal{G} - (x_i, x_{i'})$ is still structurally left invertible for all feasible sets of attack nodes and step 4 is feasible. □

Theorem 19 shows we can obtain a minimal network resilient to perfect attacks even with constraints on communication. While Algorithm 2 gives a method to construct such an optimal communication network, the method and complexity of this approach is unclear. Nonetheless, if we can compute a maximum set of vertex disjoint paths from a vertex x_i to o, we can determine outgoing neighbors of agent x_i which should not be deleted. In particular, we should keep edges from x_i to p neighbors through which there exists p disjoint paths to o, with the condition that none of these paths should contain x_i as an intermediate vertex. We use Dinic's algorithm in Algorithm 3 to solve Problem 28. The complexity of this algorithm is $O(n|\mathcal{V}_H|^{\frac{1}{2}}|\mathcal{E}_H|)$ where $|\mathcal{V}_H| = 2|\mathcal{V}|,\ |\mathcal{E}_H| = |\mathcal{E}'| + |\mathcal{V}| - 1$.

B. Joint Constrained Optimization of Sensing and Communication

In the previous section we assumed that we are given a set of observed agents and then decide to optimize the network. In general, we can again consider the problem of jointly optimizing sensing and communication. As before, we minimize an affine function of the number of communication links and number of sensors. Now, however, we consider constraints on the matrix $[A]$ encoded through $[\bar{A}]$. The resulting optimization problem is given below.

$$\begin{aligned}
&\min_{[A],[C],m} \alpha_1 \|A\|_0 + \alpha_2 m \\
&|S_i| \geq p,\ [A]_{ii} \neq 0,\ \|C^i\|_0 \leq 1, \quad i = 1, \cdots, n \\
&C \in \mathbb{R}^{m \times n},\ \|C_j\|_0 = 1, \quad j = 1, \cdots, m \\
&[\bar{A}]_{uv} = 0 \implies [A]_{uv} = 0,\ m \in \{p, p+1, \cdots, n\}. \qquad (29)
\end{aligned}$$

The problem is equivalent to

$$\begin{aligned}
&\min_{m,[C]} \left(\min_{[A]} \alpha_1 \|A\|_0(m) + \alpha_2 m \right), \\
&= \min_{p^* \leq m \leq n} \alpha_1 n(p+1) + (\alpha_2 - \alpha_1) m.
\end{aligned}$$

where p^* is the fewest number of observers needed to ensure a system is not perfectly attackable. If $\alpha_1 < \alpha_2$, so that network links are more expensive, we take the maximum number of sensors, $m^* = n$, thus assigning a sensor to each agent. If $\alpha_2 > \alpha_1$, so that sensors are more costly than network links, we simply take the minimum number of sensors, $m^* = p^*$. Once we obtain a feasible $[C]$ with p^* observers, we solve problem 28 for fixed $[C]$ using Algorithm 3.

In practice, determining p^* or a feasible matrix $[C]$ is not trivial. To determine that a set of k sensors can not be used to design systems which are not perfectly attackable, $\binom{n}{k}$ possible configurations must be checked. Nonetheless, in the case that $p = 1$, we can efficiently determine the number of observers needed using strongly connected components.

1) Strongly Connected Component Decomposition: We first consider the graph $\mathcal{G}^{\mathcal{X}} = (\mathcal{X}, \mathcal{E}_{\mathcal{X},X})$ obtained by removing all observers and only considering the structural system associated with $[\bar{A}]$. The digraph $\mathcal{G}^{\mathcal{X}}$ is strongly connected if there is a path between any pair of vertices. Moreover, a strongly connected component (SCC) is a maximum subgraph of $\mathcal{G}^{\mathcal{X}}$, that is strongly connected.

It is noted that any digraph can be uniquely decomposed into disjoint SCCs. Moreover, we can represent such a decomposition using a directed acyclic graph (DAG), that is, a graph without cycles [28]. A supernode in such a graph corresponds to a single SCC and there exists a directed edge between two SCCs if and only if there exists an edge between vertices belonging to the corresponding SCCs. We say that an SCC is non-bottom linked if there is no outgoing directed edge from that SCC to another SCC. Otherwise it is bottom linked. Let $\mathcal{G}^{\mathcal{X}}_S = (\mathcal{V}_S, \mathcal{E}_S)$ denote the DAG obtained from the SCC decomposition of $\mathcal{G}^{\mathcal{X}}$. We can obtain the DAG in $O(|\mathcal{X}| + |\mathcal{E}_{\mathcal{X},X}|)$ time complexity [29].

2) Case p = 1: Given the SCC decomposition of $\mathcal{G}^{\mathcal{X}}$ we can characterize the number of observers needed to ensure structural left invertibility when the defender must be resilient to $p = 1$ attackers. In particular we have the following result.

Theorem 20. *The minimum number of observers needed for $[\bar{A}]$ to be structurally left invertible for all feasible attack policies when $p = 1$ is given by the number of non-bottom linked SCCs in $\mathcal{G}^{\mathcal{X}}_S$.*

Proof. To avoid perfect attacks when $p = 1$, there must exist at least one directed path from every node to an observer. We first argue that each non-bottom linked SCC requires one unique observer. Suppose instead that a non-bottom linked SCC $X_1 \in \mathcal{V}_S$ does not have an observer. Let $x_i \in X_1$. There must be a directed path from x_i to an observer. However, since X_1 has no outgoing edges to another SCC, such a path can not exist. Thus, the number of non-bottom linked SCCs in $\mathcal{G}^{\mathcal{X}}_S$ is a lower bound on the number of observers needed.

We next show that there exists a system which is not perfectly attackable with a number of observers equal to the number of non-bottom linked SCCs in $\mathcal{G}_S^{\mathcal{X}}$. To do this, we arbitrarily assign an observer to each non-bottom linked SCC. Suppose x_i is in a non-bottom linked SCC. Since the SCC is strongly connected, there exists a path from x_i to an observer. Suppose instead that x_i is in a bottom linked SCC. We observe that in $\mathcal{G}_S^{\mathcal{X}}$, there must exist a path from a bottom linked SCC $X_j \in \mathcal{V}_S$ to a non-bottom linked SCC $X_l \in \mathcal{V}_S$. If not, $\mathcal{G}_S^{\mathcal{X}}$ contains a cycle. However, by construction [28], $\mathcal{G}_S^{\mathcal{X}}$ is an acyclic graph. Thus, there exists a path from a bottom linked SCC to a non-bottom linked SCC. This implies that there exists a directed path from x_i to an observer. As a result, the system is not perfectly attackable. □

In the case that $p = 1$, the above theorem states that the fewest number of observers needed is equal to the number of non-bottom linked SCCs. Moreover, with regards to the joint optimization of sensors and communication in Problem 29, if sensing is more expensive than communication, then for $p = 1$, we can simply and efficiently obtain a SCC decomposition of $[\bar{A}]$. We can next assign observers to each non-bottom linked SCC. Finally, we would solve the constrained optimization problem 28 for fixed $[C]$ using Algorithm 3. As mentioned previously, robust design to ensure that a system is not perfectly attackable can also ensure system observability.

Corollary 21. *Suppose there exist self loops in every state vertex. The structural left invertibility of* $([A], [B_F^a], [C], [D_F^a])$ *for all possible sets of malicious attackers* $F \in \mathcal{F}$ *implies the structural observability of* $([A], [C])$.

Proof. Using Theorem 3 of [30], a system with self loops in every state vertex is structurally observable if and only if every non-bottom linked SCC of the associated system digraph has an observed edge. Thus, from Theorem 20, structural observability is equivalent to structural left invertibility for all sets of attackers $\mathcal{F}$ when $p = 1$. If a system is structurally left invertible in the presence $p > 1$ attackers, then it is structurally left invertible with 1 attacker. The result follows. □

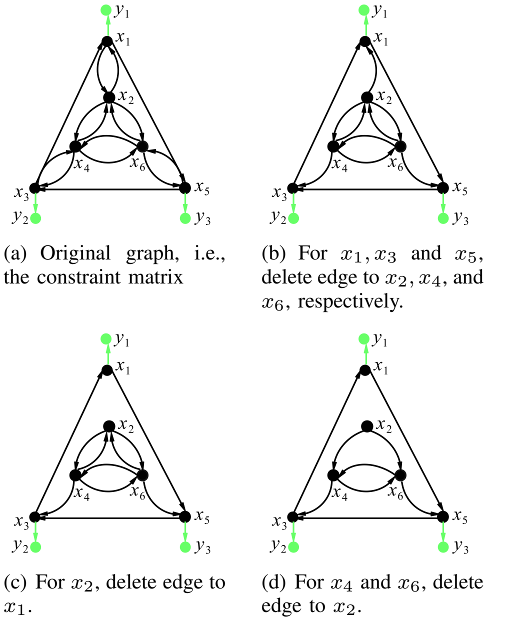
Consequently, when we perform robust design to prevent perfect attacks, we also satisfy traditional objectives by making the DCS structurally observable. A structurally observable system is also observable for almost all feasible (A, C).

VII. Example and Simulation

In Section VII-A, we provide an illustrative example which shows how we obtain the solution of Problem 28 based on Algorithm 3. In [19] we considered system design in a vehicle platoon. In Section VII-B, we now consider an example of distributed formation control of multi-agent systems [4], [5], [31]. Specifically, we analyze how the scale and connectivity of the network influences the runtime of Algorithm 3.

A. Illustrative Example

Consider a 9-state system measured by 3 sensors, as depicted in Fig. 1. The graphical representation of the constraint matrix $[\bar{A}]$ is depicted in Fig. 1(a) with self loops abstracted away. If $[\bar{A}]_{uv}$ is not a fixed zero, there exists an edge (x_v, x_u). Suppose the goal is to design an optimal communication network which ensures robustness under at most $p = 2$ attacks. Recalling Algorithm 3, we start with the digraph associated with $[\bar{A}]$, and for each of the state vertices x_i we keep p outgoing neighbors which ensure the size of the minimum vertex separator between (x_i, o) is p. Fig. 1(b)-1(d) shows the results of these iterations.



(a) Original graph, i.e., the constraint matrix

(b) For x_1, x_3 and x_5, delete edge to x_2, x_4, and x_6, respectively.

(c) For x_2, delete edge to x_1.

(d) For x_4 and x_6, delete edge to x_2.

Fig. 1. Process of Algorithm 3, starting with the constraint matrix in (a).

B. Numerical Example: Distributed Formation Control

Consider a multi-agent system with n agents, where the agents are able to locally communicate with each other. The goal of formation control could be organizing the agents according to certain 2-D formations. In this case, the state of the system corresponds to the location of agents in a two-dimensional space. In the simulation, we generated a $n \times 2$ matrix of random variables under uniform distribution $U[0, 1]$, which represent the initial location of n agents.

n	r	$\|[\bar{A}]\|_0$	p	m	$\|A\|_0^*$	Runtime (sec)
100	0.15	632	2	10	190	425.58
100	0.2	980	2	10	190	776.31
100	0.3	2020	2	10	190	1766.97
100	0.2	970	3	15	285	768.49
100	0.2	938	4	20	380	682.13
50	0.2	182	2	10	90	25.11
150	0.2	2386	2	10	290	1.1430×10^4

TABLE I
Runtime of Algorithm 3 for different n, p, r parameters

Due to communication cost and noise, the communications between agents are restricted to a certain radius r. As a result, we can compute the constraint matrix $[\bar{A}]$ by enumerating the distance between every pair of agents. More precisely, if the distance between the i-th agent and j-th agent is less than r, then $[\bar{A}]_{ij} = [\bar{A}]_{ji} = 1$. Otherwise, $[\bar{A}]_{ij} = [\bar{A}]_{ji} = 0$. Under such a constraint matrix, the goal is to design a minimum communication network $\bar{A}$, which is resistant to p attacks.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCNS.2016.2573741, IEEE Transactions on Control of Network Systems

To generate $[C]$, we apply graph clustering [32] to the graph associated with $[\bar{A}]$ and group the vertices into five clusters. In each cluster, we assign p sensors to p arbitrary state vertices.

Table I lists the simulation results, where we consider different values of n, p and r, and record the runtime of Algorithm 3 using a Macbook Pro running Ubuntu Linux with a 2.7 GHz Intel Core i5 processor. In order to compute p essential neighbors of x_i, corresponding to step 5-6 of Algorithm 3, toolbox TOMLAB/CPLEX [33] was incorporated. From the table we can tell that the most influential factor with respect to runtime is the scale of the network, i.e., n. The parameter r, which influences the connectivity of the network, also significantly influences the speed of the algorithm.

VIII. Conclusion

In this article, we considered the problem of securely designing DCS to prevent a special class of integrity attacks known as perfect attacks where an attacker can manipulate the state without affecting the measurements of the system. Previous work considered structural left invertibility to characterize the ability to avoid perfect attacks from a fixed set of malicious nodes. However, in this article, we used vertex separators to characterize resilience of the system to perfect attacks from all sets of feasible malicious nodes and efficiently verify the robustness of networks to perfect attacks. We then considered the goal of designing systems which are not perfectly attackable while simultaneously minimizing communication in the system. Such a design ensures deterministic detection of attacks. We examined the joint design of sensor placement and communication with constraints on the network, arriving at polynomial time algorithms to obtain feasible and minimal network realizations. In future work, we would like to consider a version of the problem of minimal design when every communication link has a unique cost.

References

[1] F. Blaabjerg, R. Teodorescu, M. Liserre, and A. V. Timbus, "Overview of control and grid synchronization for distributed power generation systems," *IEEE Transactions in Industrial Electronics*, vol. 53, no. 5, pp. 1398–1409, 2006.

[2] S. M. Amin and B. F. Wollenberg, "Toward a smart grid: power delivery for the 21st century," *Power and Energy Magazine*, vol. 3, no. 5, pp. 34–41, 2005.

[3] B. Sinopoli, C. Sharp, L. Schenato, S. Schaffert, and S. S. Sastry, "Distributed control applications within sensor networks," *Proceedings of the IEEE*, vol. 91, no. 8, pp. 1235–1246, 2003.

[4] R. Olfati-Saber and R. M. Murray, "Distributed cooperative control of multiple vehicle formations using structural potential functions," in *15th IFAC World Congress*, 2002, pp. 346–352.

[5] W. Ren and R. W. Beard, *Distributed consensus in multi-vehicle cooperative control*. Springer-Verlag, 2008.

[6] R. Olfati-Saber and R. M. Murray, "Consensus and cooperation in networked multi-agent systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 215–233, 2007.

[7] A. A. Cardenas, S. Amin, and S. Sastry, "Secure control: Towards survivable cyber-physical systems," in *The 28th International Conference on Distributed Computing Systems Workshops*, 2008, pp. 495–500.

[8] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *Critical Infrastructure Protection*. Springer US, 2008, pp. 73–82.

[9] Y. Liu, M. Reiter, and P. Ning, "False data injection attacks against state estimation in electric power grids," in *Proceedings of the 16th ACM Conference on Computer and Communications Security*, Chicago, IL, 2009.

[10] H. Sandberg, A. Teixeira, and K. H. Johansson, "On security indices for state estimators in power networks," in *First Workshop on Secure Control Systems*, Stockholm, Sweden, 2010.

[11] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *First Workshop on Secure Control Systems*, Stockholm, Sweden, 2010.

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

[13] S. Sundaram and C. Hadjicostis, "Distributed function calculation via linear iterative strategies in the presence of malicious agents," *IEEE Transactions on Automatic Control*, vol. 56, no. 7, pp. 1495–1508, 2011.

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

[15] S. Sundaram, M. Pajic, C. Hadjicostis, R. Mangharam, and G. J. Pappas, "The wireless control network: Monitoring for malicious behavior," in *49th IEEE Conference on Decision and Control*, Atlanta, GA, 2010, pp. 5979–5984.

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

[17] H. Cam, P. Mouallem, Y. Mo, B. Sinopoli, and B. Nkrumah, "Modeling impact of attacks, recovery, and attackability conditions for situational awareness," in *2014 IEEE International Inter-Disciplinary Conference on Cognitive Methods in Situation Awareness and Decision Support (CogSIMA)*, San Antonio, Texas, 2014, pp. 181–187.

[18] F. Pasqualetti, F. Dorfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[19] S. Weerakkody, X. Liu, S. H. Son, and B. Sinopoli, "A graph theoretic characterization of perfect attackability and detection in distributed control systems," in *To appear: 2016 American Control Conference*, 2015. [Online]. Available: http://arxiv.org/abs/1510.04712

[20] C. Miller and C. Valasek, "Remote exploitation of an unaltered passenger vehicle," Tech. Rep., August 2015. [Online]. Available: http://illmatics.com/Remote%20Car%20Hacking.pdf

[21] Y. Nakahira and Y. Mo, "Dynamic state estimation in the presence of compromised sensory data," in *54th IEEE Conference on Decision and Control*, Osaka, Japan, 2015, pp. 5808 – 5813.

[22] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas, "Robustness of attack-resilient state estimators," in *ICCPS'14: ACM/IEEE 5th International Conference on Cyber-Physical Systems (with CPS Week 2014)*. IEEE Computer Society, 2014, pp. 163–174.

[23] C. T. Lin, "Structural controllability," *IEEE Transactions on Automatic Control*, vol. 19, no. 3, pp. 201–208, 1974.

[24] K. Menger, "Zur allgemeinen kurventheorie," *Fundamenta Mathematicae*, vol. 1, no. 10, pp. 96–115, 1927.

[25] L. R. Ford and D. R. Fulkerson, *Flows in Networks*. Princeton, New Jersey: Princeton University Press, 1962.

[26] E. A. Dinic, "An algorithm for the solution of the max-flow problem with the polynomial estimation," *Doklady Akademii Nauk*, vol. 194, no. 4, pp. 1277–1280, 1970.

[27] S. Even and R. E. Tarjan, "Network flow and testing graph connectivity," *SIAM journal on computing*, vol. 4, no. 4, pp. 507–518, 1975.

[28] X. Liu, S. Pequito, S. Kar, B. Sinopoli, and A. P. Aguiar, "Minimum sensor placement for robust observability of structured complex networks," *IEEE Transactions on Network Science and Engineering, under review*, 2015. [Online]. Available: http://arxiv.org/pdf/1507.07205v1.pdf

[29] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson, *Introduction to Algorthms*, 2nd ed. McGraw-Hill Higher Education, 2001.

[30] S. Pequito, S. Kar, and A. Aguiar, "A framework for structural input/output and control configuration selection in large-scale systems," *IEEE Transactions on Automatic Control*, vol. 61, no. 2, pp. 303 – 318, 2015.

[31] K. Yeom, "Distributed formation control for communication relay with positionless flying agents," *Multimedia, Computer Graphics and Broadcasting*, pp. 18–27, 2012.

[32] S. E. Schaeffer, "Survey: Graph clustering," *Comput. Sci. Rev.*, vol. 1, no. 1, pp. 27–64, Aug. 2007. [Online]. Available: http://dx.doi.org/10.1016/j.cosrev.2007.05.001

[33] "TOMLAB optimization: TOMLAB /CPLEX." [Online]. Available: http://tomopt.com/tomlab/products/cplex/

Sean Weerakkody received the B.S degree in Electical Engineering and Mathematics from the University of Maryland, College Park, USA, in 2012. He was awarded the National Defense Science and Engineering Graduate fellowship in 2014. He is currently currently pursuing the Ph.D. degree in Electrical and Computer Engineering at Carnegie Mellon University, Pittsburgh, PA.

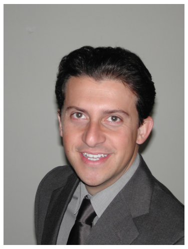
His research interests include secure design and active detection in cyber-physical systems and estimation in sensor networks.

Xiaofei Liu received the B.Sc. degree from Department of Automation, Tsinghua University, Beijing, China, in 2012 and the M.Sc. degree from Department of Electrical and Computer Engineering, Carnegie Mellon University, Pittsburgh, PA, in 2015. She is currently working towards the Ph.D. degree in Electrical and Computer Engineering at Carnegie Mellon University, Pittsburgh, PA.

Her research interests include analysis and secure design of networked control systems based on structural system theory.

Sang Hyuk Son is the Dean of the Graduate School and the Director of CPS Global Center at DGIST. He has been a Professor of Computer Science Department at the University of Virginia, and WCU Chair Professor at Sogang University. He received the B.S. degree in electronics engineering from Seoul National University, M.S. degree from KAIST, and the Ph.D. in computer science from University of Maryland, College Park. He has been a Visiting Professor at KAIST, City University of Hong Kong, Ecole Centrale de Lille in France, and Linkoping University and University of Skovde in Sweden.

Prof. Son is IEEE Fellow, and a member of both the Korean Academy of Science & Technology and the National Academy of Engineering of Korea. He is serving as an Associate Editor for the ACM Transactions on Cyber Physical Systems, and has served on the editorial board of the IEEE Transactions on Computers, the IEEE Transactions on Parallel and Distributed Systems, and the Real-Time Systems Journal. He is a founding member of the ACM/IEEE CPS Week, and serving as a member of the steering committee for the IEEE RTCSA, Cyber Physical Systems Week, and SEUS. He received the Outstanding Contribution Award from the Cyber Physical Systems Week in 2012. His research interests include cyber physical systems, real-time and embedded systems, database and data services, and wireless sensor networks. He has written or co-authored over 300 papers and edited/authored four books in these areas. According to Google Scholar, his papers received over 11,000 citations total. His research has been funded by the Korean Government, National Research Foundation, USA National Science Foundation, DARPA, Office of Naval Research, Department of Energy, National Security Agency, and IBM.



Bruno Sinopoli received the Dr. Eng. degree from the University of Padova in 1998 and his M.S. and Ph.D. in Electrical Engineering from the University of California at Berkeley, in 2003 and 2005 respectively. After a postdoctoral position at Stanford University, Dr. Sinopoli joined the faculty at Carnegie Mellon University where he is an associate professor in the Department of Electrical and Computer Engineering with courtesy appointments in Mechanical Engineering and in the Robotics Institute and co-director of the Smart Infrastructure Institute, a research center aimed at advancing innovation in the modeling analysis and design of smart infrastructure. Dr. Sinopoli was awarded the 2006 Eli Jury Award for outstanding research achievement in the areas of systems, communications, control and signal processing at U.C. Berkeley, the 2010 George Tallman Ladd Research Award from Carnegie Mellon University and the NSF Career award in 2010. His research interests include the modeling, analysis and design of Secure by Design Cyber-Physical Systems with applications to Energy Systems, Interdependent Infrastructures and Internet of Things.