The Wireless Control Network: Monitoring for Malicious Behavior

Shreyas Sundaram, Miroslav Pajic, Christoforos N. Hadjicostis, Rahul Mangharam, and George J. Pappas

***Abstract*— We consider the problem of stabilizing a plant with a network of resource constrained wireless nodes. In a companion paper, we developed a protocol where each node repeatedly transmits a linear combination of the values in its neighborhood. For certain topologies, we showed that these linear combinations can be designed so that the closed loop system is stable (i.e., the wireless network itself acts as a controller for the plant). In this paper, we design a Intrusion Detection System (IDS) for this control scheme, which observes the transmissions of certain nodes in the network and uses that information to (a) recover the plant outputs (for data-logging and diagnostic purposes) and (b) identify malicious behavior by any of the wireless nodes in the network. We show that if the connectivity of the network is sufficiently high, the IDS only needs to observe a subset of the nodes in the network in order to achieve this objective. Our approach provides a characterization of the set of nodes that should be observed, a systematic procedure for the IDS to use to identify the malicious nodes and recover the outputs of the plant, and an upper bound on the delay required to obtain the necessary information.**

I. Introduction

Industrial control systems are often deployed in large, spatially distributed plants that involve numerous sensors, actuators and internal process variables. Interconnecting the various components of these systems has traditionally been achieved through physical wiring, which is often difficult to do (when the plant contains hard-to-reach or dangerous areas), expensive, and fault-prone. However, the advent of low-cost and reliable wireless networks promises to alleviate many of these issues [1], [2]. With this technology, sensor measurements of plant variables can be transmitted to controllers, data centers and plant operators without the need for excessive wiring, thereby yielding gains in efficiency and profitability for the operator.

The topic of control over networks (wireless or otherwise) has been intensively studied by researchers over the past decade, leading to design procedures for controllers that are tolerant to network imperfections such as packet dropouts and transmission delays [3], [4], [5]. These works typically adopt the convention of having a dedicated controller/estimator located somewhere in the network, and study the stability of the closed loop system assuming that the sensor-estimator and/or controller-actuator communication channels are unreliable (dropping packets with a certain probability, for example). In the companion paper [6], we introduced the *Wireless Control Network*, a new paradigm for control over a wireless network where the network *itself* acts as the controller (instead of having a specially designated node performing this task). Specifically, we considered a wireless network consisting of simple nodes that are able to exchange information only with their direct neighbors. We devised a protocol where each node transmits, at each time-step, a single value that is a linear combination of the values in its neighborhood. Nodes that have access to the outputs of the plant (i.e., those nodes that are located near the plant sensors) include those measurements in their updates, and the plant actuators apply a linear combination of the transmissions of nodes that are closest to them. This novel protocol effectively causes the wireless network to behave as a linear system with sparsity constraints on the system matrices (corresponding to the topology of the network). We provided a numerical design procedure (based on linear matrix inequalities) to determine the appropriate linear combinations for each node to use in order to stabilize the plant, even when packets are dropped with a (sufficiently low) probability. As discussed in [6], this scheme has several benefits over traditional approaches to designing networked control systems:

- It can explicitly incorporate very simple (computationally constrained) nodes into the design procedure.
- It simplifies the transmission scheduling polices for the network.
- It can easily handle practical scenarios involving large-scale plants that have multiple (geographically dispersed) sensing and actuation points.

While the stability of networked control systems under benign packet-drop scenarios has been well studied, the need for a rigorous theory of *security* in industrial control systems has only recently started to gain attention [7], [8], [9], [10], [11]. In domains such as chemical process industries, aviation and critical infrastructure, attacks on the control systems could have disastrous consequences. The report [12] makes several key recommendations for "designing-in" security into industrial control systems. One of the points highlighted by the report is the need to maintain accurate logs of the plant and controller behavior, and to analyze the information contained in those logs in order to quickly detect and isolate anomalies. In traditional (data) networks, this type of monitoring is performed with an *Intrusion Detection System* (IDS), which essentially raises an alarm if the observed traffic flow in the network deviates from expected patterns [13]. The application of IDSs to wireless networks is a relatively new area of research [14], and the

This research has been supported by NSF CNS-0931239.

S. Sundaram, M. Pajic. R. Mangharam and G. J. Pappas are with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA 19014. Email: {ssund, pajic, rahulm, pappasg}@seas.upenn.edu. C. N. Hadjicostis is with the Department of Electrical and Computer Engineering, University of Cyprus. Email: chadjic@ucy.ac.cy.

paper [15] suggests an IDS for wireless networks in process control industries. The design in [15] captures (at a policy level) attacks such as jamming, flooding the network with large numbers of packets, and corruptions in the formatting of data transmitted by certain nodes.

A more dangerous (and difficult to detect) attack in control networks is that of data *modification*, where malicious nodes subtly change the contents of messages that they are passing through the network, but otherwise follow the normal rules of transmission. In this paper, we describe how to design an IDS to detect data modification attacks in the control scheme proposed in [6]. The IDS will be responsible for observing the transmissions of certain nodes in the network in order to (a) recover the outputs of the plant (e.g., for fault-diagnosis purposes), and (b) detect and identify data modification attacks by nodes in the network; the overall architecture of the plant, control network and IDS is shown in Fig. 1. We show that the wireless control scheme from [6] allows malicious behavior to be identified by examining the transmissions of only a *subset* of the nodes in the network, provided that the network topology satisfies certain conditions. We provide an explicit characterization of the subset of nodes that needs to be monitored, along with a procedure for the IDS to follow in order to extract the required information from the transmissions of those nodes.

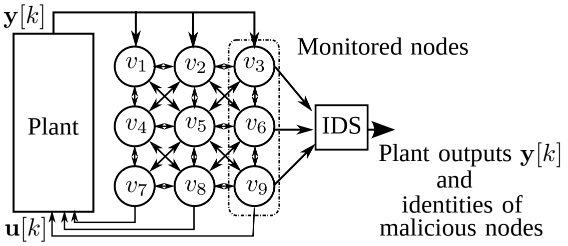

Fig. 1. Architecture of the wireless control network with an IDS.

II. Notation and Background on Graph Theory

We use $\mathbf{e}_i$ to denote the column vector (of appropriate size) with a 1 in its i-th position and 0's elsewhere, and the symbol $\mathbf{1}$ to denote the column vector (of appropriate size) consisting of all 1's. The symbol $\mathbf{I}_N$ denotes the $N \times N$ identity matrix, and $\mathbf{A}'$ indicates the transpose of matrix $\mathbf{A}$. The cardinality of a set $\mathcal{S}$ is denoted by $|\mathcal{S}|$, and for two sets $\mathcal{S}$ and $\mathcal{R}$, we use $\mathcal{S} \setminus \mathcal{R}$ to denote the set of elements in $\mathcal{S}$ that are not in $\mathcal{R}$. The set of nonnegative integers is denoted by $\mathbb{N}$.

A graph is an ordered pair $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is a set of vertices (or nodes), and $\mathcal{E}$ is a set of ordered pairs of different vertices, called directed edges. The vertices in the set $\mathcal{N}_{v_i} = \{v_j | (v_j, v_i) \in \mathcal{E}\}$ are said to be neighbors of vertex v_i. A *subgraph* of $\mathcal{G}$ is a graph $\mathcal{H} = \{\bar{\mathcal{V}}, \bar{\mathcal{E}}\}$, with $\bar{\mathcal{V}} \subseteq \mathcal{V}$ and $\bar{\mathcal{E}} \subseteq \mathcal{E}$ (where all edges in $\bar{\mathcal{E}}$ are between vertices in $\bar{\mathcal{V}}$).

A *path* P from vertex v_{i_0} to vertex v_{i_t} is a sequence of vertices $v_{i_0} v_{i_1} \cdots v_{i_t}$ such that $(v_{i_j}, v_{i_{j+1}}) \in \mathcal{E}$ for $0 \leq j \leq t-1$. The nonnegative integer t is the *length* of the path. We will call a graph *disconnected* if there exists at least one pair of vertices $v_i, v_j \in \mathcal{V}$ such that there is no path from v_j to v_i. The *connectivity* of the graph is defined as the smallest number of vertices that must be removed to disconnect the graph, and is denoted by κ. A set of paths $P_1, P_2, \ldots, P_r$ are vertex disjoint if no vertex appears in more than one path. Given two subsets $\mathcal{V}_1, \mathcal{V}_2 \subset \mathcal{V}$, an *r-linking* from $\mathcal{V}_1$ to $\mathcal{V}_2$ is a set of r vertex disjoint paths, each with start vertex in $\mathcal{V}_1$ and end vertex in $\mathcal{V}_2$. Note that if $\mathcal{V}_1$ and $\mathcal{V}_2$ are not disjoint, we will take their common vertices to be vertex disjoint paths between $\mathcal{V}_1$ and $\mathcal{V}_2$ of length zero. The following classical result will play a role in our derivations (e.g., see [16]).

Lemma 1: Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ have connectivity κ, and let $\mathcal{V}_1$ and $\mathcal{V}_2$ be subsets of $\mathcal{V}$, each of size at least κ. Then there is a κ-linking from $\mathcal{V}_1$ to $\mathcal{V}_2$ (and vice versa).

III. The Wireless Control Network

Consider a plant of the form

$$\begin{aligned} \mathbf{x}[k+1] &= \mathbf{A}\mathbf{x}[k] + \mathbf{B}\mathbf{u}[k] \\ \mathbf{y}[k] &= \mathbf{C}\mathbf{x}[k], \end{aligned} \tag{1}$$

with $\mathbf{A} \in \mathbb{R}^{n \times n}, \mathbf{B} \in \mathbb{R}^{n \times m}$ and $\mathbf{C} \in \mathbb{R}^{p \times n}$. The output vector $\mathbf{y}[k] = \begin{bmatrix} y_1[k] & y_2[k] & \ldots & y_p[k] \end{bmatrix}'$ contains measurements of the plant state vector $\mathbf{x}[k]$ provided by the sensors $s_1, \ldots, s_p$. The input vector $\mathbf{u}[k] = \begin{bmatrix} u_1[k] & u_2[k] & \ldots & u_m[k] \end{bmatrix}'$ corresponds to the signals applied to the plant by actuators $a_1, \ldots, a_m$.

The plant is to be controlled using a wireless network consisting of a set of nodes that interact with each other and with the sensors and actuators installed on the plant. Each node in the network is equipped with a radio transceiver along with (limited) memory and computational capabilities.[1] Similarly, each sensor and actuator on the plant contains a radio transceiver, allowing them to communicate with neighboring nodes. The wireless network is described by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of N nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the radio connectivity (communication topology) in the network (i.e., edge $(v_j, v_i) \in \mathcal{E}$ if node v_i can receive information directly from node v_j). In addition, we define $\mathcal{V}_S \subset \mathcal{V}$ as the set of nodes that can receive information directly from at least one sensor, and $\mathcal{V}_A \subset \mathcal{V}$ as the set of nodes whose transmissions can be heard by at least one actuator. We will refer to $\mathcal{V}_S$ as the *source* nodes in the network. In this paper, we will also assume that there are some *malicious nodes* in the network, given by the set $\mathcal{F} \subset \mathcal{V}$. These malicious nodes will transmit false values (perhaps by conspiring with each other) in an attempt to damage the system in some way. Note that the set $\mathcal{F}$ is unknown *a priori*.

In our development, we will find it convenient to consider a new graph $\bar{\mathcal{G}}$ that captures how the plant outputs enter into the wireless control network. This graph is obtained by taking the graph of the network $\mathcal{G}$ and adding p new vertices

[1] We will model these resource constraints by limiting the state maintained by each node to be a scalar. As discussed in [6], the control scheme can also be applied to cases where nodes are allowed to maintain state vectors.

$\mathcal{S} = \{s_1, s_2, \dots, s_p\}$, corresponding to the sensors on the plant. Define the edge set

$$\mathcal{E}_I = \left\{ \; (s_l, v_j) \; \middle| \; \begin{array}{c} s_l \in \mathcal{S}, v_j \in \mathcal{V}_S, \\ s_l\text{'s value is available to node } v_j \end{array} \right\}.$$

We then obtain $\bar{\mathcal{G}} = \{\mathcal{V} \cup \mathcal{S}, \mathcal{E} \cup \mathcal{E}_I\}$.

The proposed control scheme (introduced in [6], [17]) consists of having each node in the network update its value to be a linear combination of its previous value and the values of its neighbors. In addition, each source node will include a linear combination of the sensor measurements (i.e., plant outputs) that it receives at each time-step. Finally, the malicious nodes will update their values arbitrarily at each time-step. Mathematically, if we let $z_i[k]$ denote node v_i's value at time-step k, we obtain the update equations:[2]

$$z_i[k+1] = \tag{2}$$
$$\begin{cases} w_{ii}z_i[k] + \sum_{v_j \in \mathcal{N}_{v_i}} w_{ij}z_j[k] \\ \qquad + \sum_{s_j \in \mathcal{N}_{v_i}} h_{ij}y_j[k] & \text{if } v_i \in \mathcal{V}_S \setminus \mathcal{F}, \\ w_{ii}z_i[k] + \sum_{v_j \in \mathcal{N}_{v_i}} w_{ij}z_j[k] \\ \qquad + \sum_{s_j \in \mathcal{N}_{v_i}} h_{ij}y_j[k] + f_i[k] & \text{if } v_i \in \mathcal{V}_S \cap \mathcal{F}, \\ w_{ii}z_i[k] + \sum_{v_j \in \mathcal{N}_{v_i}} w_{ij}z_j[k] + f_i[k] & \text{if } v_i \in \mathcal{F} \setminus \mathcal{V}_S, \\ w_{ii}z_i[k] + \sum_{v_j \in \mathcal{N}_{v_i}} w_{ij}z_j[k] & \text{if } v_i \notin \mathcal{V}_S \cup \mathcal{F}. \end{cases}$$

The scalars w_{ij} and h_{ij} specify the linear combinations that are computed by each node in the network. The scalar $f_i[k]$ is an additive error[3] committed by node v_i at time-step k if it is malicious. If we let $\mathcal{F} = \{v_{j_1}, v_{j_2}, \dots, v_{j_{|\mathcal{F}|}}\}$ denote the set of malicious nodes, and aggregate the values transmitted by all nodes at time-step k into the value vector $\mathbf{z}[k] = \begin{bmatrix} z_1[k] & z_2[k] & \cdots & z_N[k] \end{bmatrix}'$, the transmission strategy for the entire system can be represented as

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{W}\mathbf{z}[k] + \mathbf{H}\mathbf{y}[k] \\ &\quad + \underbrace{\begin{bmatrix} \mathbf{e}_{j_1} & \mathbf{e}_{j_2} & \cdots & \mathbf{e}_{j_{|\mathcal{F}|}} \end{bmatrix}}_{\mathbf{E}_{\mathcal{F}}} \underbrace{\begin{bmatrix} f_{j_1}[k] \\ f_{j_2}[k] \\ \vdots \\ f_{j_{|\mathcal{F}|}}[k] \end{bmatrix}}_{\mathbf{f}[k]} \\ &= \mathbf{W}\mathbf{z}[k] + \underbrace{\begin{bmatrix} \mathbf{H} & \mathbf{E}_{\mathcal{F}} \end{bmatrix}}_{\mathbf{B}_{\mathcal{F}}} \underbrace{\begin{bmatrix} \mathbf{y}[k] \\ \mathbf{f}[k] \end{bmatrix}}_{\mathbf{v}[k]} , \end{aligned} \tag{3}$$

for all $k \in \mathbb{N}$. In the above equation, the (i, j) entry of $\mathbf{W}$ satisfies $w_{ij} = 0$ if $v_j \notin \mathcal{N}_{v_i}$, and the (i, j) entry of $\mathbf{H}$ satisfies $h_{ij} = 0$ if $s_j \notin \mathcal{N}_{v_i}$. We assume that $\mathbf{z}[0]$ (i.e., the initial state of the wireless control network) is known to the IDS. Recall that the symbol $\mathbf{e}_i$ denotes a vector with a single 1 in the i–th position and zeros elsewhere.

At each actuator $l \in \{1, 2, \dots, m\}$, we apply the input $u_l[k] = \mathbf{g}_l\mathbf{z}[k]$, where $\mathbf{g}_l$ is a vector that specifies a linear combination of the values transmitted by the nodes $\mathcal{V}_A$ that are near that actuator.[4] The update strategy for the network can therefore be represented as

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{W}\mathbf{z}[k] + \mathbf{H}\mathbf{y}[k] + \mathbf{E}_{\mathcal{F}}\mathbf{f}[k] \\ \mathbf{u}[k] &= \mathbf{G}\mathbf{z}[k], \end{aligned}$$

where the matrices $\mathbf{W} \in \mathbb{R}^{N \times N}$, $\mathbf{H} \in \mathbb{R}^{N \times p}$ and $\mathbf{G} \in \mathbb{R}^{m \times N}$ have sparsity constraints determined by the underlying network topology. When there are no malicious nodes (i.e., $\mathcal{F} = \emptyset$), the overall closed loop system evolves as:

$$\begin{bmatrix} \mathbf{x}[k+1] \\ \mathbf{z}[k+1] \end{bmatrix} = \left[\begin{array}{cc} \mathbf{A} & \mathbf{BG} \\ \mathbf{HC} & \mathbf{W} \end{array} \right] \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix} \triangleq \hat{\mathbf{A}} \begin{bmatrix} \mathbf{x}[k] \\ \mathbf{z}[k] \end{bmatrix}.$$

Matrix $\hat{\mathbf{A}}$ is *structured*, in that certain entries are forced to be zero (corresponding to the topology of the wireless control network). Let Ψ_s denote the set of all tuples $(\mathbf{W}, \mathbf{H}, \mathbf{G})$ that satisfy the required sparsity patterns and that cause the matrix $\hat{\mathbf{A}}$ to have all its eigenvalues inside the open unit circle. In [6], [17], a numerical procedure was provided to find an element of Ψ_s (if one exists).

In this paper, we consider the problem of data collection and analysis in this network for the purpose of identifying malicious behavior by a nonempty subset $\mathcal{F}$ of nodes. Specifically, we will describe the design of an *Intrusion Detection System*,[5] whose task is to collect data from the network in order to (a) recover the plant outputs[6] $\mathbf{y}[k]$ and (b) detect and isolate anomalous behavior in the wireless control network. Clearly, one trivial option would be for the IDS to simply listen to the transmissions of *every* node and sensor in the network, and double-check that all nodes are indeed computing the proper linear combinations at each time-step. However, this is not a satisfactory solution, since the entire point of the wireless control network is to avoid the communication infrastructure required for a centralized solution of this kind. Instead, we would like a way to identify the malicious nodes in the network and obtain the plant outputs by viewing the transmissions of just a *subset* $\mathcal{T} \subset \mathcal{V}$ of the nodes. Perhaps surprisingly, we will show that this is possible with an appropriate choice of the set $\mathcal{T}$ (provided that the network topology satisfies certain conditions), even though the transmissions of the nodes have been designed specifically with the goal of plant stabilization in mind. In other words, the above design for the wireless control network *simultaneously* achieves the dual objectives of stabilizing the plant *and* providing the IDS with enough information to diagnose failures and malicious

[2] The neighborhood $\mathcal{N}_v$ of a vertex v is with respect to the graph $\bar{\mathcal{G}}$.

[3] This model allows a malicious node to update and transmit an arbitrary value by choosing the error term $f_i[k]$ appropriately. It also captures the scenario where multiple malicious nodes update their values in a coordinated manner. We assume that malicious nodes cannot send conflicting values to different neighbors, due to the broadcast nature of the communications.

[4] In this work, we do not consider the possibility of malicious actuators that apply arbitrary inputs to the system. Such behavior can potentially be identified by using the outputs of the plant $\mathbf{y}[k]$ and applying appropriate fault-diagnosis techniques, as described below.

[5] We assume that this is a trusted entity, with sufficient computational and storage capabilities to analyze the data that it receives from the network.

[6] This information can be used by the IDS for tasks such as diagnosing faults that occur within the plant (e.g., using the techniques described in [18]). Since this is a rather general problem, we will not delve into the details of *how* the IDS uses the outputs $\mathbf{y}[k]$ further in this paper, and instead, will concentrate on ensuring that the IDS can obtain these outputs, in addition to identifying malicious nodes in the control network.

behavior. Broadly speaking, our analysis will reveal that if the connectivity of the wireless control network is at least $p+2f$, and if each sensor measurement is heard by at least $p+2f$ nodes, then the IDS can deduce the above information from the transmissions of *any* $p+2f$ nodes in the network, as long as there are no more than f malicious nodes during any D contiguous time-steps (where D is an integer that we will characterize later).

Remark 1: In this work, we will not consider a probabilistic drop model for the channels between nodes; the possibility of a large number of (accidental) packet losses incurred by such a model complicates the task of isolating malicious behavior, and future research will be devoted to addressing this more general scenario. However, our model does capture the case where there is a limited (bounded) number of packet-dropping channels in any set of D contiguous time-steps. Specifically, note that a dropped packet from node v_j to v_i can be modeled as "malicious" behavior by node v_i, where the additive error $f_i[k]$ is selected to cancel out the contribution of $z_j[k]$ in v_i's update. Thus, one can effectively trade an actual malicious node for a dropped packet in our analysis, as long as the total number of actual malicious nodes and dropped packets in any set of D contiguous time-steps is less than or equal to f. □

IV. Analysis Algorithm for the Intrusion Detection System

For any set $\mathcal{T} \subset \mathcal{V}$, denote the vector of transmissions of the nodes in that set at time-step k by $\mathbf{t}[k]$. We can write

$$\mathbf{t}[k] = \mathbf{T}\mathbf{z}[k] \ , \tag{4}$$

where $\mathbf{T}$ is a $|\mathcal{T}| \times N$ matrix with a single 1 in each row capturing the positions of the vector $\mathbf{z}[k]$ that are in the set $\mathcal{T}$, and zeros elsewhere. In this section, we provide a procedure for the IDS to use to parse the values $\mathbf{t}[k]$, $k \in \mathbb{N}$, in order to recover the plant outputs $\mathbf{y}[k]$, and identify anomalous behavior by any nodes in the network.

In our development, we will find it useful to consider a slightly more general version of the system model (3). For any subset $\mathcal{Q} = \{v_{q_1}, v_{q_2}, \ldots, v_{q_{|\mathcal{Q}|}}\} \subset \mathcal{V}$ of nodes, let $\mathbf{E}_{\mathcal{Q}} = \begin{bmatrix}\mathbf{e}_{q_1} & \mathbf{e}_{q_2} & \cdots & \mathbf{e}_{q_{|\mathcal{Q}|}}\end{bmatrix}$, and define $\mathbf{B}_{\mathcal{Q}} = \begin{bmatrix}\mathbf{H} & \mathbf{E}_{\mathcal{Q}}\end{bmatrix}$ (where $\mathbf{H}$ is the matrix from (3) specifying the linear combinations of the plant outputs that are used by the source nodes). Note that $\mathbf{B}_{\mathcal{Q}}$ has $p+|\mathcal{Q}|$ columns. The values seen by the IDS over $L+1$ time-steps (for some nonnegative integer L) for the system

$$\begin{aligned}\mathbf{z}[k+1] &= \mathbf{W}\mathbf{z}[k] + \mathbf{B}_{\mathcal{Q}}\mathbf{v}[k] \\ \mathbf{t}[k] &= \mathbf{T}\mathbf{z}[k]\end{aligned} \tag{5}$$

are given by

$$\underbrace{\begin{bmatrix}\mathbf{t}[k] \\ \mathbf{t}[k+1] \\ \mathbf{t}[k+2] \\ \vdots \\ \mathbf{t}[k+L]\end{bmatrix}}_{\mathbf{t}[k:k+L]} = \underbrace{\begin{bmatrix}\mathbf{T} \\ \mathbf{TW} \\ \mathbf{TW}^2 \\ \vdots \\ \mathbf{TW}^L\end{bmatrix}}_{\boldsymbol{\Theta}_L} \mathbf{z}[k] + \mathbf{M}_L^{\mathcal{Q}} \underbrace{\begin{bmatrix}\mathbf{v}[k] \\ \mathbf{v}[k+1] \\ \mathbf{v}[k+2] \\ \vdots \\ \mathbf{v}[k+L-1]\end{bmatrix}}_{\mathbf{v}[k:k+L-1]}, \tag{6}$$

where

$$\mathbf{M}_L^{\mathcal{Q}} \triangleq \begin{bmatrix}\mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{TB}_{\mathcal{Q}} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{TWB}_{\mathcal{Q}} & \mathbf{TB}_{\mathcal{Q}} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{TW}^{L-1}\mathbf{B}_{\mathcal{Q}} & \mathbf{TW}^{L-2}\mathbf{B}_{\mathcal{Q}} & \cdots & \mathbf{TB}_{\mathcal{Q}}\end{bmatrix}. \tag{7}$$

When $L = N-1$, $\boldsymbol{\Theta}_L$ is the *observability matrix* for the pair $(\mathbf{W}, \mathbf{T})$, and we will call $\mathbf{M}_L^{\mathcal{Q}}$ the *input matrix corresponding to the set* $\mathcal{Q}$.

The following theorem shows that the IDS can recover the desired quantities from the transmissions of nodes in $\mathcal{T}$, provided that a certain algebraic condition holds. We will later relate this algebraic condition to conditions on the network topology and choices of the monitored nodes $\mathcal{T}$.

Theorem 1: Suppose that there exists an integer D such that, for all possible sets $\mathcal{Q}$ of $2f$ nodes, the matrix $\mathbf{M}_D^{\mathcal{Q}}$ satisfies

$$\text{rank}\left(\mathbf{M}_D^{\mathcal{Q}}\right) = p + |\mathcal{Q}| + \text{rank}\left(\mathbf{M}_{D-1}^{\mathcal{Q}}\right) \ . \tag{8}$$

Then, as long as there are no more than f malicious nodes in the network during any set of D contiguous time-steps, the IDS can uniquely recover the plant outputs $\mathbf{y}[k]$ and identify all of the malicious nodes with a delay of D time-steps, based on the transmissions of the nodes in $\mathcal{T}$. □

Before proceeding with the proof of the above theorem, we provide a more detailed explanation of condition (8). Specifically, note from (7) that for any set $\mathcal{Q}$, the last $(L-1)$ block-columns of $\mathbf{M}_L^{\mathcal{Q}}$ have the form $\begin{bmatrix}\mathbf{0} \\ \mathbf{M}_{L-1}^{\mathcal{Q}}\end{bmatrix}$, and thus have rank equal to the rank of $\mathbf{M}_{L-1}^{\mathcal{Q}}$. Condition (8) is therefore equivalent to saying that the first $p+|\mathcal{Q}|$ columns of $\mathbf{M}_D^{\mathcal{Q}}$ must be linearly independent of each other, and of all other columns in $\mathbf{M}_D^{\mathcal{Q}}$. With this interpretation in hand, we are now ready to continue with the proof of Theorem 1.

Proof: [Theorem 1] Consider time-steps $k = 0, 1, \ldots, D$, and suppose that the malicious nodes during this period are a subset of the set $\mathcal{F} = \{v_{j_1}, v_{j_2}, \ldots, v_{j_f}\}$. From (3), (4) and (6), the values seen by the IDS over these time-steps are given by

$$\mathbf{t}[0:D] = \boldsymbol{\Theta}_D\mathbf{z}[0] + \mathbf{M}_D^{\mathcal{F}}\mathbf{v}[0:D-1] \ , \tag{9}$$

where $\mathbf{v}[k] = \begin{bmatrix}\mathbf{y}'[k] & \mathbf{f}'[k]\end{bmatrix}'$. Note that the IDS knows the quantities $\mathbf{t}[0:D]$ and $\boldsymbol{\Theta}_D\mathbf{z}[0]$, but it does not know the set $\mathcal{F}$ or the values $\mathbf{v}[0:D-1]$. The IDS will try to identify these unknown parameters based on the known quantities.

Let $\mathcal{F}_1, \mathcal{F}_2, \ldots, \mathcal{F}_{\binom{N}{f}} \subset \mathcal{V}$ denote all possible sets of f nodes, and let $\mathbf{M}_D^{\mathcal{F}_1}, \mathbf{M}_D^{\mathcal{F}_2}, \ldots, \mathbf{M}_D^{\mathcal{F}_{\binom{N}{f}}}$ denote the input matrices corresponding to these sets. With these matrices in hand, suppose that the IDS finds the first $j \in \{1, 2, \ldots, \binom{N}{f}\}$ such that the vector $\mathbf{t}[0:D] - \boldsymbol{\Theta}_D\mathbf{z}[0]$ is in the column space of the matrix $\mathbf{M}_D^{\mathcal{F}_j}$. This means that the IDS can find a vector $\bar{\mathbf{v}}[0:D-1]$ such that

$$\mathbf{M}_D^{\mathcal{F}_j}\bar{\mathbf{v}}[0:D-1] = \mathbf{t}[0:D] - \boldsymbol{\Theta}_D\mathbf{z}[0].$$

The vector $\bar{\mathbf{v}}[0:D-1]$ is the IDS's *estimate* of the value of $\mathbf{v}[0:D-1]$ (note that the value $\bar{\mathbf{v}}[k] = \begin{bmatrix}\bar{\mathbf{y}}'[k] & \bar{\mathbf{f}}'[k]\end{bmatrix}'$ contains estimates of the plant outputs and the malicious errors at time-step k). Substituting (9) into the above expression and rearranging, we have

$$\mathbf{M}_D^{\mathcal{F}}\mathbf{v}[0:D-1] - \mathbf{M}_D^{\mathcal{F}_j}\bar{\mathbf{v}}[0:D-1] = \mathbf{0} \ .$$

Let $\{\mathcal{F}, \mathcal{F}_j\}$ denote the set that is obtained by concatenating sets $\mathcal{F}$ and $\mathcal{F}_j$ (i.e., it is the union of the two sets, with duplications allowed). Exploiting the form of matrix $\mathbf{M}_D^{\mathcal{Q}}$ shown in (7), the above expression can be written as

$$\underbrace{\begin{bmatrix} \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{T}\mathbf{B}_{\{\mathcal{F},\mathcal{F}_j\}} & \cdots & \mathbf{0} \\ \mathbf{T}\mathbf{W}\mathbf{B}_{\{\mathcal{F},\mathcal{F}_j\}} & \cdots & \mathbf{0} \\ \vdots & \ddots & \vdots \\ \mathbf{T}\mathbf{W}^{D-1}\mathbf{B}_{\{\mathcal{F},\mathcal{F}_j\}} & \cdots & \mathbf{T}\mathbf{B}_{\{\mathcal{F},\mathcal{F}_j\}} \end{bmatrix}}_{\mathbf{M}_D^{\{\mathcal{F},\mathcal{F}_j\}}} \begin{bmatrix} \tilde{\mathbf{v}}[0] \\ \tilde{\mathbf{v}}[1] \\ \tilde{\mathbf{v}}[2] \\ \vdots \\ \tilde{\mathbf{v}}[D-1] \end{bmatrix} = \mathbf{0} \tag{10}$$

where $\mathbf{B}_{\{\mathcal{F},\mathcal{F}_j\}} = \begin{bmatrix}\mathbf{H} & \mathbf{E}_{\mathcal{F}} & \mathbf{E}_{\mathcal{F}_j}\end{bmatrix}$ and

$$\tilde{\mathbf{v}}[k] = \begin{bmatrix} \mathbf{y}[k] - \bar{\mathbf{y}}[k] \\ \mathbf{f}[k] \\ -\bar{\mathbf{f}}[k] \end{bmatrix} \ .$$

Now consider the matrix $\mathbf{M}_D^{\mathcal{F}\cup\mathcal{F}_j}$. Since $\mathcal{F}\cup\mathcal{F}_j$ has at most $2f$ nodes, equation (8) in the statement of the theorem indicates that the first $p + |\mathcal{F}\cup\mathcal{F}_j|$ columns of the matrix $\mathbf{M}_D^{\mathcal{F}\cup\mathcal{F}_j}$ are linearly independent of each other, and of all other columns of the matrix. Now, note that the matrix $\mathbf{M}_D^{\{\mathcal{F},\mathcal{F}_j\}}$ is obtained from matrix $\mathbf{M}_D^{\mathcal{F}\cup\mathcal{F}_j}$ simply by duplicating certain columns (namely, the columns corresponding to nodes that appear in both $\mathcal{F}$ and $\mathcal{F}_j$). Consider a node $v_l \in \mathcal{F}$. If $v_l \notin \mathcal{F}_j$, then the column corresponding to v_l within the first $p+2f$ columns of $\mathbf{M}_D^{\{\mathcal{F},\mathcal{F}_j\}}$ will be linearly independent of all other columns in $\mathbf{M}_D^{\{\mathcal{F},\mathcal{F}_j\}}$ (since this column will also appear in the first $p+|\mathcal{F}\cup\mathcal{F}_j|$ columns of $\mathbf{M}_D^{\mathcal{F}\cup\mathcal{F}_j}$). This means that equation (10) can be satisfied only if $f_l[0] = 0$. On the other hand, if $f_l[0] \neq 0$, the only way for equation (10) to be satisfied is if $v_l \in \mathcal{F}_j$ and $\bar{f}_l[0] = f_l[0]$. In other words, if equation (8) is satisfied, any malicious node that commits an error during the first time-step will appear in set $\mathcal{F}_j$, and its additive error can be found by the IDS.

Next, note from (8) that the first p columns of $\mathbf{M}_D^{\{\mathcal{F},\mathcal{F}_j\}}$ will be linearly independent of each other and of all other columns in that matrix (since these columns also appear in $\mathbf{M}_D^{\mathcal{F}\cup\mathcal{F}_j}$ and are not duplicated in $\mathbf{M}_D^{\{\mathcal{F},\mathcal{F}_j\}}$). This means that the only way for equation (10) to be satisfied is if $\bar{\mathbf{y}}[0] = \mathbf{y}[0]$. Thus, the IDS has also recovered the outputs of the plant that were injected into the network at time-step $k=0$.

At this point, the IDS knows $\mathbf{y}[0]$ and the identities of those nodes in $\mathcal{F}$ that committed errors during time-step 0, along with the exact values of their additive errors. The IDS can then use (3) to obtain the transmitted values of all nodes at time-step $k=1$ as

$$\mathbf{z}[1] = \mathbf{W}\mathbf{z}[0] + \mathbf{H}\mathbf{y}[0] + \mathbf{B}_{\mathcal{F}_j}\bar{\mathbf{f}}[0] \ .$$

Now, using the identity

$$\mathbf{t}[1:D+1] = \mathbf{\Theta}_D\mathbf{z}[1] + \mathbf{M}_D^{\mathcal{F}}\mathbf{v}[1:D] \ ,$$

the IDS can repeat the above process to find the values of $\mathbf{y}[1]$ along with the identities of the nodes that are malicious during time-step $k=1$. By repeating the above procedure for all positive values of k, the IDS can obtain the identities of all malicious nodes and the errors that they commit, along with the source streams $\mathbf{y}[k]$ for all k, simply by listening to the transmissions of the nodes in $\mathcal{T}$. ■

Remark 2: It is worth noting that the decoding procedure specified in the above proof requires the testing of up to $\binom{N}{f}$ matrices (in the worst case) in order to locate the malicious nodes. If one assumes that the set of malicious nodes does not change over time, then at time-step k, the IDS can restrict its search to only those sets $\mathcal{F}_j$ that contain all of the malicious nodes from time-steps less than k. This reduces the computational burden on the IDS in subsequent time-steps. However, if we allow the IDS to repeat the search for malicious nodes at each time-step, this analysis procedure is also able to tolerate cases where the set of malicious nodes changes over time (with the only constraint being that no more than f nodes are malicious during any set of D contiguous time-steps). The development of a more efficient method to parse the transmissions of the monitored nodes is an important venue for future research. □

V. Network Topology Conditions for Misbehavior Identification and Data Recovery

Theorem 1 provides a decoding procedure for the IDS provided that condition (8) is true. In this section, we will use results from the theory of *dynamic system inversion* and *structured linear systems* to relate this condition to conditions on the network topology.

A. System Inversion

Consider the wireless control network given by equations (3) and (4). The quantities $\mathbf{y}[k]$ and $\mathbf{f}[k]$ in (3) are unknown to the IDS, and so linear systems of this type are termed *linear systems with unknown inputs*[7] in the control literature (e.g., see [19]). For such systems, it is often of interest to "invert" the system in order to reconstruct some or all of the unknown inputs, and this problem has been studied under the moniker of *dynamic system inversion*. We will now summarize some pertinent results from the literature on system inversion, and apply them to the problem of detecting and identifying malicious nodes in the wireless control network.

For any set $\mathcal{Q} \subseteq \mathcal{V}$, the output of the linear system (5) over $L+1$ time-steps (for some nonnegative integer L) is given by (6). Alternatively, we can consider the transfer function

$$\mathbf{P}(z) = \mathbf{T}\left(z\mathbf{I} - \mathbf{W}\right)^{-1}\mathbf{B}_{\mathcal{Q}} \ ,$$

which is a $|\mathcal{T}| \times (p + |\mathcal{Q}|)$ matrix of rational functions of z.

[7] In our case, the set $\mathcal{F}$ (and thus the matrix $\mathbf{B}_{\mathcal{F}}$) is also unknown to the IDS, so the system given by (3) and (4) is more general than the linear systems with unknown inputs commonly considered in the literature.

Definition 1: The system (5) is said to have an L-delay inverse if there exists a system with transfer function $\widehat{\mathbf{P}}(z)$ such that $\widehat{\mathbf{P}}(z)\mathbf{P}(z) = z^{-L}\mathbf{I}_{p+|\mathcal{Q}|}$. The system is invertible if it has an L-delay inverse for some finite L. The least integer L for which an L-delay inverse exists is called the inherent delay of the system. □

In order for the system to be invertible, its transfer function must have rank $p+|\mathcal{Q}|$ over the field of rational functions in z. The following result follows directly from [19] and [20] (which studied the problem of dynamic system inversion) and provides a test for invertibility in terms of the system matrices $\mathbf{W}, \mathbf{B}_{\mathcal{Q}}$ and $\mathbf{T}$.

Theorem 2 ([19], [20]): For any nonnegative integer L,

$$\text{rank}(\mathbf{M}_L^{\mathcal{Q}}) \leq p + |\mathcal{Q}| + \text{rank}(\mathbf{M}_{L-1}^{\mathcal{Q}}) \tag{11}$$

with equality if and only if the system has an L-delay inverse (note that $\text{rank}(M_{-1}^{\mathcal{Q}})$ is defined to be zero). If the system is invertible, its inherent delay will not exceed $L = N - p - |\mathcal{Q}| + 1$. □

Note that condition (11) means that the first $p + |\mathcal{Q}|$ columns of $\mathbf{M}_L^{\mathcal{Q}}$ must be linearly independent of each other, and of all other columns in $\mathbf{M}_L^{\mathcal{Q}}$. Taking $\mathcal{Q}$ to be any set of $2f$ nodes, this is precisely the condition that is required for us to detect and identify malicious nodes (as specified in equation (8) in Theorem 1). In other words, the problem of identifying malicious nodes in the wireless control network can be viewed as a problem of linear system inversion. Thus the task is now to find conditions on the network topology and a set of nodes $\mathcal{T}$ that will ensure that the linear system specified by the matrices $(\mathbf{W}, \mathbf{B}_{\mathcal{Q}}, \mathbf{T})$ is invertible for every choice $\mathcal{Q} \subset \mathcal{V}$ of $2f$ nodes. To solve this problem, we will first use the theory of *linear structured systems* to obtain a graph-theoretic characterization of invertibility.

B. *Structured Systems*

A linear system of the form (5) is said to be *structured* if each entry of the matrices $\mathbf{W}, \mathbf{B}_{\mathcal{Q}}$ and $\mathbf{T}$ is either a fixed zero or an independent free parameter [21]. Interestingly, such systems have certain properties that can be inferred purely from the zero/nonzero structure of the system matrices; these properties will hold for almost any choice of free parameters (i.e., the set of parameters for which the property does not hold has Lebesgue measure zero [21]), and thus these properties are called *generic*. Of particular relevance to this paper is the *generic normal rank* of the transfer function matrix of a structured system, which is the maximum rank (over the field of rational functions in z) of the transfer function matrix over all possible choices of free parameters.

To analyze structural properties of linear systems of the form (5), one associates a graph $\mathcal{H}$ with the structured set $(\mathbf{W}, \mathbf{B}_{\mathcal{Q}}, \mathbf{T})$ as follows. The vertex set of $\mathcal{H}$ is given by $\mathcal{V} \cup \mathcal{I} \cup \mathcal{O}$, where $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ is the set of state vertices, $\mathcal{I} = \{i_1, i_2, \ldots, i_{p+|\mathcal{Q}|}\}$ is the set of input vertices, and $\mathcal{O} = \{o_1, o_2, \ldots, o_{|\mathcal{T}|}\}$ is the set of output vertices. The edge set of $\mathcal{H}$ is given by $\mathcal{E}_{vv} \cup \mathcal{E}_{iv} \cup \mathcal{E}_{vo}$, where $\mathcal{E}_{vv} = \{(v_j, v_l) \mid \mathbf{W}_{lj} \neq 0\}$, $\mathcal{E}_{iv} = \{(i_j, v_l) \mid \mathbf{B}_{\mathcal{Q},lj} \neq 0\}$, and $\mathcal{E}_{vo} = \{(v_j, o_l) \mid \mathbf{T}_{lj} \neq 0\}$ (where $\mathbf{W}_{lj}$ indicates entry (l, j) of matrix $\mathbf{W}$, and so forth). The following theorem characterizes the generic normal rank of the transfer function of a structured linear system in terms of the graph $\mathcal{H}$.

Theorem 3 ([21], [22]): Let the graph of a structured linear system be given by $\mathcal{H}$. Then the generic normal rank of the transfer function of the system is equal to the maximal size of a linking in $\mathcal{H}$ from $\mathcal{I}$ to $\mathcal{O}$. □

The above result says that if the graph of the structured system (5) has $p + |\mathcal{Q}|$ vertex disjoint paths from the inputs to the outputs, then for almost any choice of free parameters in $\mathbf{W}$, $\mathbf{B}_{\mathcal{Q}}$ and $\mathbf{T}$, the transfer function matrix $\mathbf{T}(z\mathbf{I} - \mathbf{W})^{-1}\mathbf{B}_{\mathcal{Q}}$ will have full column rank. Based on Theorem 2, this will mean that the first $p + |\mathcal{Q}|$ columns of the matrix $\mathbf{M}_{N-p-|\mathcal{Q}|+1}^{\mathcal{Q}}$ will be linearly independent of all other columns in $\mathbf{M}_{N-p-|\mathcal{Q}|+1}^{\mathcal{Q}}$.

We now have a graph-theoretic characterization of the invertibility of linear structured systems, and are in place to apply this to the problem of identifying malicious behavior and recovering the plant outputs in the wireless control network.

C. *Topological Conditions for Identifying Malicious Nodes*

From Theorem 1 and Theorem 2, the IDS can identify up to f malicious nodes if the linear system given by the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{Q}}, \mathbf{T})$ is invertible for every set $\mathcal{Q} \subset \mathcal{V}$ of up to $2f$ nodes. To verify that this property holds, note that for any given set $\mathcal{Q}$, the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{Q}}, \mathbf{T})$ essentially defines a structured linear system, with the only exception being that the nonzero entries in the matrices $\mathbf{E}_{\mathcal{Q}}$ (where $\mathbf{B}_{\mathcal{Q}} = \begin{bmatrix}\mathbf{H} & \mathbf{E}_{\mathcal{Q}}\end{bmatrix}$) and $\mathbf{T}$ are taken to be "1", rather than free parameters. However, this is of no consequence, since each nonzero entry in those matrices appears in a row and column by itself, and thus can essentially be "scaled" to a free parameter by an appropriate redefinition of the inputs and outputs (e.g., see [23]). Thus, we can proceed with applying the above results on structured system theory to the tuple $(\mathbf{W}, \mathbf{B}_{\mathcal{Q}}, \mathbf{T})$, which brings us to the following result.

Theorem 4: Let $\bar{\mathcal{G}} = \{\mathcal{V} \cup \mathcal{S}, \mathcal{E} \cup \mathcal{E}_I\}$ denote the graph of the wireless control network $\mathcal{G}$ augmented with the sensor vertices $\mathcal{S}$ and the corresponding edges. Let $\mathcal{T} \subset \mathcal{V}$ denote the set of monitored nodes. Suppose that for every possible set $\mathcal{Q} \subset \mathcal{V}$ of $2f$ nodes, the graph $\bar{\mathcal{G}}$ contains a $(p + 2f)$–linking from $\mathcal{S} \cup \mathcal{Q}$ to $\mathcal{T}$. Then, for almost any element $(\mathbf{W}, \mathbf{H}, \mathbf{G}) \in \Psi_s$ (if it is nonempty), there exists an integer $D \leq N - p - 2f + 1$ such that the IDS can recover the outputs of the plant and identify all malicious nodes with a delay of at most D time-steps, as long as there are no more than f malicious nodes in any set of D contiguous time-steps. □

Proof: For any set $\mathcal{Q} \subset \mathcal{V}$ of $2f$ nodes, consider the graph[8] $\mathcal{H}_{\mathcal{Q}}$ associated with the structured set $(\mathbf{W}, \mathbf{B}_{\mathcal{Q}}, \mathbf{T})$. To obtain this graph, start by taking the graph of the network $\mathcal{G}$ (which captures the vertices and interconnections in the matrix $\mathbf{W}$). To this graph, add $p+2f$ input vertices (denoted

[8] The notation $\mathcal{H}_{\mathcal{Q}}$ is used to denote the fact that this graph is associated with the structured set $(\mathbf{W}, \mathbf{B}_{\mathcal{Q}}, \mathbf{T})$, for a particular set $\mathcal{Q}$ of $2f$ nodes.

by $\mathcal{I}$) which will connect to the nodes in the graph according to the structure of the input matrix $\mathbf{B}_\mathcal{Q}$. Specifically, p of these input vertices correspond to the plant sensors $\mathcal{S}$ (which produce $\mathbf{y}[k]$), and each of these has outgoing edges to the nodes in $\mathcal{V}_S$ (specified by the structure of matrix $\mathbf{H}$). The other $2f$ input vertices each have a single outgoing edge to a node in $\mathcal{Q}$ (corresponding to the single 1 in each column of $\mathbf{E}_\mathcal{Q}$). Next, add $|\mathcal{T}|$ output vertices (denoted by the set $\mathcal{O}$), and place a single edge from each node in the set $\mathcal{T}$ to a node in $\mathcal{O}$, corresponding to the single nonzero entry in each row of the matrix $\mathbf{T}$. Furthermore, add a self loop to every state vertex corresponding to the nonzero entries on the diagonal of the matrix $\mathbf{W}$.

From the statement of the theorem, note that graph $\bar{\mathcal{G}}$ contains a linking of size $p + 2f$ from $\mathcal{S} \cup \mathcal{Q}$ to $\mathcal{T}$, for any set $\mathcal{Q}$ of $2f$ nodes. This linking also exists in the graph $\mathcal{H}_\mathcal{Q}$, since $\bar{\mathcal{G}}$ is a subgraph of $\mathcal{H}_\mathcal{Q}$.[9] This linking can be extended to a linking from the entire set $\mathcal{I}$ to $\mathcal{T}$ in $\mathcal{H}_\mathcal{Q}$ simply by including the edges from the set $\mathcal{I} \setminus \mathcal{S}$ to the set $\mathcal{Q}$. Finally, this linking can be further extended to a linking from $\mathcal{I}$ to $\mathcal{O}$ simply by including the edges from each vertex in $\mathcal{T}$ to the corresponding output vertex in $\mathcal{O}$. From Theorem 3, we see that the system $(\mathbf{W}, \mathbf{B}_\mathcal{Q}, \mathbf{T})$ will be invertible for almost any choice of matrices $\mathbf{W}$ and $\mathbf{H}$ (subject to the required sparsity patterns). This genericness implies that invertibility will hold simultaneously for all of the sets $(\mathbf{W}, \mathbf{B}_\mathcal{Q}, \mathbf{T})$ for every set $\mathcal{Q}$ of $2f$ nodes with almost any choice of free parameters in the matrices $\mathbf{W}$ and $\mathbf{H}$. From Theorem 2, the first $p + |\mathcal{Q}|$ columns of the matrix $\mathbf{M}^\mathcal{Q}_{N-p-2f+1}$ will be linearly independent of each other and of all other columns in $\mathbf{M}^\mathcal{Q}_{N-p-2f+1}$. Thus, condition (8) in Theorem 1 is satisfied, and the IDS can uniquely determine the identities of the malicious nodes, as well as the values of the plant outputs, based on the transmissions of the nodes in $\mathcal{T}$, with a delay of at most $N - p - 2f + 1$ time-steps.

Finally, we show that there is a tuple $(\mathbf{W}, \mathbf{H}, \mathbf{G})$ in the set Ψ_s (which contains all stabilizing structured matrices for the plant and is assumed to be nonempty) that allows the IDS to recover the desired information. This is easily done by noting that the set of matrices for which the system is stable has nonzero measure in the space $\mathbb{R}^r$ (where r is the number of free parameters in the matrices $\mathbf{W}$ and $\mathbf{H}$). More precisely, if we let $\lambda \in \mathbb{R}^r$ denote a numerical vector of free parameters in $\mathbf{W}$ and $\mathbf{H}$ that produces stability (e.g., obtained from the design procedure in [6], [17]), the closed loop system will remain stable for any parameter vectors λ^* satisfying the component-wise inequalities $\lambda - \epsilon\mathbf{1} \leq \lambda^* \leq \lambda + \epsilon\mathbf{1}$, for sufficiently small $\epsilon > 0$; this is because the eigenvalues of a matrix vary continuously with the parameters in that matrix. Thus, the set of parameters for which the system is stable has measure at least $(2\epsilon)^r > 0$, whereas the set of parameters for which the system is not invertible has measure zero. Thus, for almost any tuple $(\mathbf{W}, \mathbf{H}, \mathbf{G}) \in \Psi_s$, the system is stable *and* allows the IDS to recover the plant outputs and identify malicious behavior. ■

Theorem 4 characterizes the set of nodes $\mathcal{T}$ that the IDS should observe in order to achieve its objectives. Specifically, $\mathcal{T}$ should be sufficiently well connected to the rest of the network (i.e., there should be enough vertex disjoint paths from the other nodes in the network to the nodes in $\mathcal{T}$). However, the fact that the theorem is framed in terms of *all possible* sets $\mathcal{Q}$ of $2f$ nodes makes it somewhat unwieldy. One can come up with a more compact condition when the entire network is sufficiently well connected, as follows.

Corollary 1: Suppose that the network $\mathcal{G}$ has connectivity at least $p+2f$, and that each sensor in $\mathcal{S}$ connects to at least $p+2f$ source nodes. Let $\mathcal{T} \subseteq \mathcal{V}$ be any set of at least $p+2f$ nodes. Then, for almost any element $(\mathbf{W}, \mathbf{H}, \mathbf{G}) \in \Psi_s$ (if it is nonempty), there exists an integer $D \leq N - p - 2f + 1$ such that the IDS can recover the outputs of the plant and identify all malicious nodes with a delay of D time-steps, as long as there are no more than f malicious nodes in any set of D contiguous time-steps.

Proof: Since the network has connectivity $p + 2f$, Lemma 1 shows that for any set $\mathcal{Q}$ of $2f$ nodes, there is a linking of size $p + 2f$ from the set $\mathcal{V}_S \cup \mathcal{Q}$ to $\mathcal{T}$ (since $|\mathcal{V}_S| \geq p + 2f$). Since each sensor in $\mathcal{S}$ connects to at least $p + 2f$ source nodes, each sensor will connect to at least p nodes in the set $\mathcal{V}_S \setminus \mathcal{Q}$. By Hall's Theorem (e.g., see [16]), there is a linking of size p from $\mathcal{S}$ to $\mathcal{V}_S \setminus \mathcal{Q}$ (this is also called a *matching*). Thus, the graph $\bar{\mathcal{G}} = \{\mathcal{V} \cup \mathcal{S}, \mathcal{E} \cup \mathcal{E}_I\}$ contains a linking of size $p + 2f$ from $\mathcal{S} \cup \mathcal{Q}$ to $\mathcal{T}$ for any set $\mathcal{Q}$ of $2f$ nodes. The conditions required for Theorem 4 are thus satisfied, from which the result follows. ■

Note that the above corollary indicates that in networks with connectivity $p + 2f$ or higher, *any* set of $p + 2f$ nodes can be chosen to be observed by the IDS in order to recover the desired information about the system. For example, consider the wireless control network shown in Fig. 1. The source nodes $\mathcal{V}_S = \{v_1, v_2, v_3\}$ have access to the plant's (scalar) output $y[k]$ at each time-step, and the plant's actuator applies a linear combination of the transmissions of the nodes from the set $\mathcal{V}_A = \{v_7, v_8, v_9\}$. Note that the connectivity of the network is $\kappa = 3$, and since there is a single sensor on the plant ($p = 1$) that connects to three nodes, Corollary 1 indicates that the IDS can detect and identify up to $f = \lfloor \frac{\kappa - p}{2} \rfloor = 1$ malicious node, simply by monitoring the transmissions of any $p + 2f$ nodes (e.g., the set $\mathcal{T} = \{v_3, v_6, v_9\}$). We will forgo a numerical example of the analysis procedure here in the interest of space, but the interested reader is directed to [17] for more details.

VI. Removing Malicious Nodes

There are various courses of action that can be taken once the IDS detects and identifies a set of malicious nodes. The most direct (and drastic) action would be to shut down the plant and dispatch appropriate personnel to physically remove the malicious nodes from the network and investigate the source of the attacks. This is clearly an option of last resort, as plant shut-downs may be expensive, time-consuming, and difficult to perform. An *online* method to

[9] Specifically, it is the graph obtained by dropping the output vertices and the $2f$ input vertices connecting to the set $\mathcal{Q}$ in $\mathcal{H}_\mathcal{Q}$.

remove the malicious nodes would be more desirable, so that the plant can continue to operate. We will now briefly describe one means of achieving this.

First, once the IDS identifies a set of malicious nodes, it broadcasts a message containing the identities of all malicious nodes to all of the nodes in the network. There are various low-overhead schemes for fault-tolerant broadcast in wireless networks (e.g., see [24]) that can be used to guarantee that each node receives the correct message. After this is done, the correct nodes in the system simply ignore the transmissions of the exposed malicious nodes. However, in order to avoid affecting the stability of the closed loop system, the computations undertaken by the malicious nodes must be migrated to other nodes in the network. This can be done by following the protocol described in the companion paper [6] for dealing with crash-failures (i.e., nodes that simply drop out of the network). In this protocol, one of the failed node's neighbors becomes a *virtual node* and assumes the role of calculating the failed node's linear combinations at each time-step. All other neighbors of the failed node increase their transmission ranges so that the virtual node will receive the same information at each time-step as the failed node did. By ignoring the malicious nodes (i.e., treating them as crash-failures) and applying the above protocol, the plant can continue to operate; note that this scheme has its limits, since nodes can only increase their transmission ranges up to a certain point, and expend greater energy in doing so. However, it provides a way for the system to gracefully degrade (and self-heal) under malicious attacks until the afflicted nodes are repaired.

VII. Summary

We considered the problem of identifying malicious behavior in a wireless control network. Under nominal conditions, each node in the network transmits (at each time-step) a linear combination of the values in its immediate neighborhood. We showed in a companion paper that the linear combination for each node can be chosen so that the transmissions of nodes closest to the actuators of the plant will be stabilizing. In this paper, we showed how to construct a IDS that observes the transmissions of just a subset of the nodes in the network, and uses that information to obtain the actual plant outputs, along with the identities of any malicious nodes. In particular, we showed that if the connectivity of the network is at least $p + 2f$, and each output of the plant is heard by at least $p + 2f$ nodes, then the IDS can recover the desired information by listening to the transmissions of *any* $p + 2f$ nodes in the network.

There are a variety of avenues for future research. First, our approach requires the IDS to consider up to $\binom{N}{f}$ matrices in order to locate the malicious nodes; a more efficient scheme for parsing the observed transmissions would reduce the computational burden on the IDS. Second, an extension of these results to the case where the channels in the network drop packets in a probabilistic manner would allow our scheme to be applied in more general (i.e., unreliable) networks.

References

[1] "Why WirelessHART?" White Paper, HART Communication Foundation, Oct. 2007.

[2] S. Amidi and A. Chernoguzov, "Wireless process control network architecture overview," White Paper, Honeywell International Inc., Mar. 2009.

[3] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, Jan. 2007.

[4] J. P. Hespanha, P. Naghshtabrizi, and Y. Xu, "A survey of recent results in networked control systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, Jan. 2007.

[5] V. Gupta, A. F. Dana, J. Hespanha, R. M. Murray, and B. Hassibi, "Data transmission over networks for estimation and control," *IEEE Transactions on Automatic Control*, vol. 54, no. 8, pp. 1807–1819, Aug. 2009.

[6] M. Pajic, S. Sundaram, J. Le Ny, R. Mangharam, and G. J. Pappas, "The wireless control network: Synthesis and robustness," in *Proc. of the 49th IEEE Conference on Decision and Control*, 2010, submitted.

[7] F. Pasqualetti, A. Bicchi, and F. Bullo, "Distributed intrusion detection for secure consensus computations," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 5594–5599.

[8] A. Giani, G. Karsai, T. Roosta, A. Shah, B. Sinopoli, and J. Wiley, "A testbed for secure and robust SCADA systems," in *Proceedings of the 14th IEEE Real-time and Embedded Technology and Applications Symposium*, 2008, pp. 1–4.

[9] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation via linear iterations in the presence of malicious agents," in *Proceedings of the American Control Conference*, 2008, pp. 1350–1361.

[10] A. A. Cárdenas, S. Amin, and S. Sastry, "Research challenges for the security of control systems," in *HOTSEC'08: Proceedings of the 3rd conference on Hot topics in security*, 2008, pp. 1–6.

[11] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *HSCC '09: Proc. of the 12th International Conference on Hybrid Systems: Computation and Control*. Berlin, Heidelberg: Springer-Verlag, 2009, pp. 31–45.

[12] K. Stouffer, J. Falco, and K. Scarfone, "Guide to industrial control systems (ICS) security," National Institute of Standards and Technology, Tech. Rep. 800-82, Sep. 2008.

[13] S. Northcutt and J. Novak, *Network Intrusion Detection*, 3rd ed. New Riders Publishing, 2003.

[14] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proc. of the 3rd IEEE Consumer Communications and Networking Conference*, 2006, pp. 640–644.

[15] T. Roosta, D. K. Nilsson, U. Lindqvist, and A. Valdes, "An intrusion detection system for wireless process control systems," in *Proceedings of the 5th IEEE International Conference on Mobile Ad Hoc and Sensor Systems*, 2008, pp. 866–872.

[16] D. B. West, *Introduction to Graph Theory*. Prentice-Hall Inc., Upper Saddle River, New Jersey, 2001.

[17] M. Pajic, S. Sundaram, R. Mangharam, and G. J. Pappas, "The Wireless Control Network," University of Pennsylvania, Tech. Rep., Mar. 2010.

[18] P. M. Frank, "Fault diagnosis in dynamic systems using analytical and knowledge-based redundancy – a survey and some new results," *Automatica*, vol. 26, no. 3, pp. 459–474, May 1990.

[19] M. K. Sain and J. L. Massey, "Invertibility of linear time-invariant dynamical systems," *IEEE Transactions on Automatic Control*, vol. AC-14, no. 2, pp. 141–149, Apr. 1969.

[20] A. S. Willsky, "On the invertibility of linear systems," *IEEE Transactions on Automatic Control*, vol. 19, no. 2, pp. 272–274, June 1974.

[21] 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, July 2003.

[22] J. W. van der Woude, "A graph-theoretic characterization for the rank of the transfer matrix of a structured system," *Mathematics of Control, Signals and Systems*, vol. 4, no. 1, pp. 33–40, Mar. 1991.

[23] S. Sundaram and C. N. Hadjicostis, "Distributed function calculation and consensus using linear iterative strategies," *IEEE Journal on Selected Areas in Communications: Special Issue on Control and Communications*, vol. 26, no. 4, pp. 650–660, May 2008.

[24] A. Pelc and D. Peleg, "Broadcasting with locally bounded Byzantine faults," *Information Processing Letters*, vol. 93, no. 3, pp. 109–115, Feb. 2005.