Secure Distributed Observers for a Class of Linear Time Invariant Systems in the Presence of Byzantine Adversaries

Aritra Mitra and Shreyas Sundaram

***Abstract*— We study the problem of distributed state estimation of a linear time-invariant (LTI) system by a network of sensors, some of which are subject to adversarial attacks. We develop a secure distributed estimation strategy subject to an f*-locally bounded Byzantine* adversary model, where a compromised node is given complete knowledge of the network and system dynamics, and allowed to arbitrarily deviate from the rules of any prescribed algorithm. Under such a threat model, we present sufficient conditions guaranteeing the success of our estimation strategy. Our method relies on the construction of a subgraph, which we call a Mode Estimation Directed Acyclic Graph (MEDAG), for each unstable and marginally stable eigenvalue of the plant. We provide a distributed algorithm for constructing a MEDAG and characterize graph topologies for which a MEDAG construction algorithm is guaranteed to succeed. In the process, we make connections with the literature on secure broadcasting. In the special case where there are no adversaries, our proposed method provides a new class of distributed observers with several appealing features. Our approach provides fundamental insights into the relationship that exists between the dynamics of the system, the measurement structure of the nodes, and the underlying graph topology.**

I. Introduction

The distributed estimation problem consists of a dynamical system (or plant) together with a network of nodes (or observers) that each aim to estimate the state of the plant using local measurements and information exchanges with neighbors. This widely studied problem is broadly tackled using two main approaches, namely: Kalman-filter based techniques, and LTI observer based techniques. The foundation for the distributed Kalman-filter based state estimation approach was established in [1],[2]. These methods rely on a two-step strategy: a Kalman filter based state estimate update rule, and a data fusion step based on average-consensus. However, a limitation of this method stems from the fact that it requires an infinite number (theoretically) of data fusion iterations between two consecutive time steps of the dynamics in order to reach average consensus. Recently, in [3], the authors propose a distributed Kalman filtering scheme for discrete-time linear systems which enables finite-time data fusion of agent measurements between two successive time steps of the dynamics. Although an improvement over the infinite-time data fusion case, their method still relies on a two-time-scale strategy.

In [4] and [5], the authors propose a scalar-gain estimator which runs on a single-time-scale. They provide sufficient conditions for stability of their estimator, and introduce the notion of "Network Tracking Capacity" (NTC), a measure of the most unstable dynamics (in terms of the 2-norm of the state matrix) which can be estimated with bounded mean-squared error. However, the tight coupling between the network topology and the plant dynamics typically limits the set of unstable eigenvalues that can be accommodated by their method without violating constraints imposed upon the range of the scalar gain parameter. An alternate approach that works under the broadest observability assumptions was proposed in [6], [7]. In these works, the authors rely on state augmentation for casting the distributed estimation problem as the problem of designing a decentralized stabilizing controller for an LTI plant, using the notion of fixed modes [8].

Recently, distributed algorithms designed for various applications such as consensus [9], [10], broadcast [11], optimization [12],[13] and fault detection [14] have been investigated from the perspective of security. The main challenge in such problems is to come up with strategies which are guaranteed to work (subject to certain conditions imposed on the underlying graph topology) under carefully crafted adversarial attacks. There is a very limited literature dealing with secure distributed state estimation (in the context that we are considering here). In [15] and [16], the authors employ a metric known as the 'belief divergence', which provides a measure of how much a received state estimate deviates from the other received estimates in the neighborhood of a given node. Based on this metric, the concerned node assigns 'trust' values to each of its neighbors. The authors in [17] employ a similar trust-based scheme for assigning consensus weights to neighbors. However, these works do not provide formal proofs of convergence, nor impose any conditions on the graph topology which guarantee success of their schemes. Furthermore, they can be shown to be vulnerable to carefully crafted attacks; we provide a detailed example illustrating the same in the appendix.

In this work, we provide a secure distributed state estimation algorithm with provable guarantees. In this context, our main contributions are as follows: (i) we develop a secure state estimation scheme where each regular (non-adversarial) node has knowledge of only the plant dynamics, its neighborhood and an upper bound on the number of adversaries in its neighborhood. On the other hand, an adversarial node is endowed with complete knowledge of the system model (network topology and plant dynamics) and is allowed to deviate arbitrarily from the prescribed algorithm. (ii) we provide a formal proof of convergence (under certain graph conditions) of the state estimates of the regular nodes to the true state of the plant regardless of the actions of the adversaries. (iii) we provide a sufficient condition on the

The authors are with the School of Electrical and Computer Engineering at Purdue University. Email: {mitra14,sundara2}@purdue.edu

graph topology which guarantees the success of our proposed technique.

II. System Model

A. Notation

A directed graph is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{1, \cdots, N\}$ is the set of nodes and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the edges. An edge from node j to node i, denoted by (j, i), implies that node j can transmit information to node i. The neighborhood of the i-th node is defined as $\mathcal{N}_i \triangleq \{j \,|\, (j, i) \in \mathcal{E}\}$. A node i is said to be an outgoing neighbor of node j if $(j, i) \in \mathcal{E}$. The notation $|\mathcal{V}|$ is used to denote the cardinality of a set $\mathcal{V}$. Throughout the rest of this paper, we use the terms 'nodes' and 'observers' interchangeably.

The set of all eigenvalues (modes) of a matrix $\mathbf{A}$ is denoted by $sp(\mathbf{A}) = \{\lambda \in \mathbb{C} \,|\, det(\mathbf{A} - \lambda \mathbf{I}) = 0\}$. The set of all marginally stable and unstable eigenvalues of a matrix $\mathbf{A}$ is denoted by $\Lambda_U(\mathbf{A}) = \{\lambda \in sp(\mathbf{A}) \,|\, |\lambda| \geq 1\}$. For a matrix $\mathbf{A}$, we use $a_{\mathbf{A}}(\lambda)$ and $g_{\mathbf{A}}(\lambda)$ to denote the algebraic and geometric multiplicities, respectively, of an eigenvalue $\lambda \in sp(\mathbf{A})$. An eigenvalue λ is said to be simple if $a_{\mathbf{A}}(\lambda) = g_{\mathbf{A}}(\lambda) = 1$.

B. Problem Formulation

Consider the following autonomous discrete-time linear dynamical system[1]

$$\mathbf{x}[k+1] = \mathbf{A}\mathbf{x}[k], \tag{1}$$

where $k \in \mathbb{N}$ is the discrete-time index, $\mathbf{x}[k] \in \mathbb{R}^n$ is the state vector and $\mathbf{A} \in \mathbb{R}^{n \times n}$ is the system matrix. The system is monitored by a network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consisting of N LTI nodes. The i-th node has access to a measurement of the state, given by

$$\mathbf{y}_i[k] = \mathbf{C}_i \mathbf{x}[k], \tag{2}$$

where $\mathbf{y}_i[k] \in \mathbb{R}^{r_i}$ and $\mathbf{C}_i \in \mathbb{R}^{r_i \times n}$. We denote $\mathbf{y}[k] = \left(\mathbf{y}_1^T[k] \cdots \mathbf{y}_N^T[k]\right)^T$, and $\mathbf{C} = \left(\mathbf{C}_1^T \cdots \mathbf{C}_N^T\right)^T$.

Each node is tasked with estimating the entire system state $\mathbf{x}[k]$. In particular, let $\hat{\mathbf{x}}_i[k]$ denote the state estimate of node i, which it updates at each time-step k based on information received from its neighbors and its local measurements (if any). We refer to the network of nodes maintaining and updating these estimates as a *distributed observer*. In accordance with the terminology established in [6], [7], consider the following definition.

Definition 1. ***(Omniscience)*** *A distributed observer is said to achieve omniscience if* $\lim_{k\to\infty} ||\hat{\mathbf{x}}_i[k] - \mathbf{x}[k]|| = 0, \forall i \in \{1, \cdots, N\}$*, i.e., the state estimate maintained by each node asymptotically converges to the true state of the plant.* □

There are various challenges in achieving omniscience. First, if the pair $(\mathbf{A}, \mathbf{C}_i)$ is not detectable for some (or all) $i \in \{1, \cdots, N\}$, then the corresponding nodes cannot estimate the true state of the plant based on their own local measurements, thereby dictating the need to exchange information with their neighbors. Second, the exchange of information is restricted by the underlying communication graph $\mathcal{G}$. In addition to the above challenges, in this paper, we allow for the possibility that certain nodes in the network are compromised by an adversary, and *do not* follow their prescribed state estimate update rule. We describe below the adversary model that we will be considering.

1) Adversary Model: We partition the set of nodes $\mathcal{V}$ into two subsets: $\mathcal{R}$ comprising of a set of *regular nodes*, and $\mathcal{A} = \mathcal{V} \setminus \mathcal{R}$ comprising of a set of *adversarial nodes*. In this work, we consider the *Byzantine fault model* introduced in the computer science literature [18]. Under such a model, an adversarial node can deviate from the rules of any prescribed algorithm in arbitrary ways, and can transmit different state estimates to different neighbors at the same time step. In addition, we allow the adversarial nodes to possess complete knowledge about the graph topology and the plant dynamics, i.e., an adversarial node knows the measurements received by the normal nodes at every time step. We endow such privileges to the adversaries with the aim of providing resilience to worst-case (potentially coordinated among nodes of $\mathcal{A}$) behavior.

It is apparent that no distributed estimation algorithm would succeed if all the nodes are adversarial. In the literature dealing with distributed fault-tolerant algorithms, it is a common assumption to assign an upper bound f to the total number of adversarial nodes in the network. This is known as the *f-total adversarial model*. However, to allow for a large number of adversaries in large scale networks, we will consider a *locally bounded* fault model, taken from [19],[20], defined as follows.

Definition 2. ***(f-local set)*** *A set $\mathcal{C} \subset \mathcal{V}$ is f-local if it contains at most f nodes in the neighborhood of the other nodes, i.e., $|\mathcal{N}_i \cap \mathcal{C}| \leq f$, $\forall i \in \mathcal{V} \setminus \mathcal{C}$.* □

Definition 3. ***(f-local adversarial model)*** *A set $\mathcal{A}$ of adversarial nodes is f-locally bounded if $\mathcal{A}$ is an f-local set.* □

We formally state the problem studied in this paper as follows.

Problem 1. ***(Secure Omniscience Achieving Problem)*** *Given a system of the form (1), a set of nodes interconnected by a graph $\mathcal{G}$, and an observation model at each node given by (2), formulate a state estimation scheme so that $\lim_{k\to\infty} ||\hat{\mathbf{x}}_i[k] - \mathbf{x}[k]|| = 0$, $\forall i \in \mathcal{R}$, regardless of the actions of any f-locally bounded set of Byzantine adversaries.*

III. Secure Distributed Estimation

In order to establish the key ideas for dealing with adversarial behavior while minimizing notational complexity, we make the following assumption in the rest of the paper.

Assumption 1. *All eigenvalues of the state transition matrix $\mathbf{A}$ are real and simple.*

[1] We omit noise terms in the dynamics for the ease of exposition (e.g., as in [6], [7]). However, it can be shown that the methods developed in this paper lead to bounded mean square estimation error in the presence of noise with bounded second moments. Further, it should be noted that our proposed technique will be equally applicable to continuous time systems, with straightforward modifications.

A direct consequence of having simple eigenvalues is that we can diagonalize $\mathbf{A}$ by using the coordinate transformation matrix $\mathbf{V} = [\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(n)}]$, where $\mathbf{v}^{(1)}, \cdots, \mathbf{v}^{(n)}$ are n linearly independent eigenvectors of $\mathbf{A}$. With $\mathbf{z}[k] = \mathbf{V}^{-1}\mathbf{x}[k]$, the dynamics (1) are transformed into the form

$$\begin{aligned} \mathbf{z}[k+1] &= \mathbf{M}\mathbf{z}[k] \\ \mathbf{y}_i[k] &= \bar{\mathbf{C}}_i\mathbf{z}[k], \quad \forall i \in \{1, \cdots, N\} \end{aligned} \tag{3}$$

where $\mathbf{M} = \mathbf{V}^{-1}\mathbf{A}\mathbf{V}$ is a diagonal matrix, and $\bar{\mathbf{C}}_i = \mathbf{C}_i\mathbf{V}$. We denote the eigenvalues of $\mathbf{M}$ (which are the same as those of $\mathbf{A}$) by $\lambda_1, \cdots, \lambda_n$. For each node *i*, we denote the detectable and undetectable eigenvalues[2] by the sets $\mathcal{O}_i$ and $\mathcal{UO}_i$, respectively. We define $\rho_i = |\mathcal{O}_i|$. Next, we introduce the notion of *source nodes*.

Definition 4. ***(Source nodes)*** *For each $\lambda_j \in \Lambda_U(\mathbf{A})$, the set of nodes that can detect λ_j is denoted by $\mathcal{S}_j$, and called the set of source nodes for λ_j.* □

Note that if each regular node in the network can accurately estimate $\mathbf{z}[k]$, then they can also estimate $\mathbf{x}[k]$ using the relation $\mathbf{x}[k] = \mathbf{V}\mathbf{z}[k]$. In view of this, we develop an estimation scheme which enables each regular node to estimate $\mathbf{z}[k]$. For each $\lambda_j \in \Lambda_U(\mathbf{A})$, our estimation scheme relies on separate strategies for nodes in $\mathcal{S}_j$, and $\mathcal{V} \setminus \mathcal{S}_j$. In particular, each node in $\mathcal{S}_j$ employs a Luenberger observer for estimating $z_j[k]$ (the component of $\mathbf{z}[k]$ corresponding to the eigenvalue λ_j), while the nodes in $\mathcal{V} \setminus \mathcal{S}_j$ rely on a secure consensus algorithm for asymptotically estimating that state.[3] Next, we discuss these ideas in detail.

The first step in the estimation process involves a common coordinate transformation given by $\mathbf{z}[k] = \mathbf{V}^{-1}\mathbf{x}[k]$, to be performed by each regular node of the graph. As this only relies on the knowledge of the system matrix $\mathbf{A}$ (which is assumed to be known by all the nodes), all of the nodes can do this in a distributed manner (e.g., by using an agreed-upon convention for ordering the eigenvalues and corresponding eigenvectors).

A. Design of Luenberger Observers

Consider a regular node i. Let $\mathcal{O}_i = \{\lambda_{n_1}, \lambda_{n_2}, \cdots, \lambda_{\rho_i}\}$ (recall $\rho_i = |\mathcal{O}_i|$) be the set of detectable eigenvalues for node i. Define $\mathbf{J}_i \triangleq diag(\lambda_{n_1}, \lambda_{n_2}, \cdots, \lambda_{\rho_i})$ and $\mathbf{z}_{\mathcal{O}_i}[k] \triangleq \left[z_{n_1}[k], z_{n_2}[k], \cdots, z_{\rho_i}[k]\right]^T$. Specifically, $\mathbf{z}_{\mathcal{O}_i}[k]$ is a collection of the components of the state vector $\mathbf{z}[k]$ corresponding to the detectable eigenvalues of node i. Let $\bar{\mathbf{c}}^i_{n_j}$ denote the column of $\bar{\mathbf{C}}_i$ corresponding to the eigenvalue $\lambda_{n_j} \in \mathcal{O}_i$. Define the observation matrix $\bar{\mathbf{C}}_{\mathcal{O}_i} \in \mathbb{R}^{r_i \times \rho_i}$ as $\bar{\mathbf{C}}_{\mathcal{O}_i} \triangleq \left[\bar{\mathbf{c}}^i_{n_1}, \bar{\mathbf{c}}^i_{n_2}, \cdots, \bar{\mathbf{c}}^i_{\rho_i}\right]$.

Remark 1. *Under Assumption 1, if λ_j is not detectable, then its corresponding column in $\bar{\mathbf{C}}_i$ is a zero vector [21]. Consequently, for each node i, the undetectable components of the state vector $\mathbf{z}[k]$ (corresponding to the eigenvalues in $\mathcal{UO}_i$) do not contribute to the measurement $\mathbf{y}_i[k]$.* □

Let $\hat{z}^i_{n_j}[k]$ denote the i-th node's estimate of component $z_{n_j}[k]$ (corresponding to the eigenvalue λ_{n_j}) of the state vector $\mathbf{z}[k]$. Define the composite estimate vector $\hat{\mathbf{z}}_{\mathcal{O}_i}[k] \triangleq \left[\hat{z}^i_{n_1}[k], \hat{z}^i_{n_2}[k], \cdots, \hat{z}^i_{\rho_i}[k]\right]^T$. Consider the following Luenberger observer at node i

$$\hat{\mathbf{z}}_{\mathcal{O}_i}[k+1] = \mathbf{J}_i\hat{\mathbf{z}}_{\mathcal{O}_i}[k] + \mathbf{L}_i\left(\mathbf{y}_i[k] - \bar{\mathbf{C}}_{\mathcal{O}_i}\hat{\mathbf{z}}_{\mathcal{O}_i}[k]\right), \tag{4}$$

where $\mathbf{L}_i \in \mathbb{R}^{\rho_i \times r_i}$ is an observation gain matrix at node i. From the definitions of $\mathbf{J}_i$ and $\bar{\mathbf{C}}_{\mathcal{O}_i}$, it follows that the pair $(\mathbf{J}_i, \bar{\mathbf{C}}_{\mathcal{O}_i})$ is detectable. Thus, $\mathbf{L}_i$ can be chosen so that $(\mathbf{J}_i - \mathbf{L}_i\bar{\mathbf{C}}_{\mathcal{O}_i})$ is Schur stable, and $\lim_{k\to\infty} ||\hat{\mathbf{z}}_{\mathcal{O}_i}[k] - \mathbf{z}_{\mathcal{O}_i}[k]|| = 0$. This leads to the following straightforward result which we shall use in our subsequent development.

Lemma 1. *Suppose Assumption 1 holds. Then, for each regular node $i \in \mathcal{R}$ and each $\lambda_j \in \mathcal{O}_i$, the observer given by equation (4) ensures that $\lim_{k\to\infty} |\hat{z}^i_j[k] - z_j[k]| = 0$.* □

Remark 2. *The above result shows that a node does not have to rely on information exchange with neighbors in order to estimate certain subsets of the state. Specifically, a node needs to talk to its neighbors for estimating only the portion of the state that is not locally detectable. The rest of the state space can be estimated using local measurements and by constructing observers of the form (4).* □

In the following section, we describe a secure consensus based strategy for estimating the portion of the state that is not locally detectable.

B. Consensus Based Secure State Estimate Update Rule

Consider the unstable (or marginally stable) eigenvalue $\lambda_j \in \mathcal{UO}_i$. For such an eigenvalue, node i has to rely on the information received from its neighbors (some of whom might be adversarial) in order to estimate $z_j[k]$. We propose a consensus based strategy which ensures that node i can estimate $z_j[k]$ asymptotically in the presence of adversaries. To this end, our proposed secure consensus algorithm requires each regular node $i \in \mathcal{V} \setminus \mathcal{S}_j$ to update its estimate of $z_j[k]$ using the following two stage filtering strategy:

1) At each time-step k, each regular node i collects the state estimates of $z_j[k]$ received from *only* those neighbors which belong to a certain subset $\mathcal{N}^i_j \subseteq \mathcal{N}_i$ (to be defined later), and ranks them from largest to smallest.
2) Node i removes the largest and smallest f estimates (i.e., removes $2f$ estimates in all), and updates its own state estimate using the following rule:

$$\hat{z}^i_j[k+1] = \lambda_j \sum_{l \in \mathcal{M}^i_j[k]} w^j_{il}[k]\hat{z}^l_j[k], \tag{5}$$

where the set $\mathcal{M}^i_j[k] \subset \mathcal{N}^i_j (\subseteq \mathcal{N}_i)$ denotes the set of nodes from which node i chooses to accept estimates of $z_j[k]$ at time instant k, after removing the f largest and f smallest estimates from $\mathcal{N}^i_j$. Also, $w^j_{il}[k]$ is the weight

[2]Given a pair $(\mathbf{A}, \mathbf{C}_i)$, an eigenvalue $\lambda \in \Lambda_U(\mathbf{A})$ is said to be detectable if $rank \begin{bmatrix} \mathbf{A} - \lambda\mathbf{I}_n \\ \mathbf{C}_i \end{bmatrix} = n$. Each stable eigenvalue of $\mathbf{A}$ is by default considered to be detectable.

[3]Our strategies will apply identically to each unstable and marginally stable eigenvalue of $\mathbf{M}$. Thus, we focus our discussion on a generic $\lambda_j \in \Lambda_U(\mathbf{A})$.

the i-th node associates with the l-th node at the k-th time instant, for the estimation of $z_j[k]$. The weights are non-negative and chosen to satisfy $\sum_{l\in\mathcal{M}_j^i[k]} w_{il}^j[k] = 1, \forall \lambda_j \in \mathcal{UO}_i$.

We refer to the above algorithm as the Local-Filtering based Secure Estimation (LFSE) algorithm. For implementing this algorithm, a regular node i needs to construct the set $\mathcal{N}_j^i$, $\forall \lambda_j \in \mathcal{UO}_i$, based on the relative positions of its neighbors (with respect to its own position) in the graph $\mathcal{G}$. We will provide the exact definition of $\mathcal{N}_j^i$, and a distributed algorithm for constructing such a set in a subsequent section.

Remark 3. *Notice that in the update rule defined by equation (5), a regular node $i \in \mathcal{V}\setminus\mathcal{S}_j$ does not use its own estimate value for the update, i.e., it assigns a zero self-weight to itself. The rationale for this is that λ_j is an unstable (or marginally stable) eigenvalue belonging to the unobservable subspace of node i (i.e., $\lambda_j \in \Lambda_U(\mathbf{M}) \cap \mathcal{UO}_i$), and hence it relies purely on the information provided by its neighbors to estimate the state corresponding to λ_j.* □

Remark 4. *The idea of disregarding the most extreme values in one's neighborhood, and using a convex combination of the rest for performing linear updates, is a strategy which has been adopted for secure distributed consensus in [9], and is termed the Weighted-Mean-Subsequence-Reduced (W-MSR) algorithm. For the secure distributed estimation problem, a regular node i needs to disregard information received from some of its neighbors not only based on their extreme nature, but also based on their relative positions (with respect to node i) in the graph. This is a fundamental difference between the strategies adopted for secure distributed consensus and secure distributed estimation. The latter can be thought of as a class of secure consensus problems, where the consensus value itself needs to follow a certain trajectory dictated by the given plant dynamics.* □

a) Summary of the Secure Estimation Scheme: We briefly summarize the secure estimation scheme as follows.

1) All nodes perform a common coordinate transformation defined by $\mathbf{z}[k] = \mathbf{V}^{-1}\mathbf{x}[k]$. Accordingly, a regular node i identifies its detectable and undetectable eigenvalues ($\mathcal{O}_i$ and $\mathcal{UO}_i$).
2) Each regular node i uses a Luenberger observer defined by equation (4), to estimate the states $\mathbf{z}_{\mathcal{O}_i}[k]$ corresponding to its detectable eigenvalues.
3) For each undetectable eigenvalue $\lambda_j \in \mathcal{UO}_i$, each regular node i follows the LFSE algorithm governed by equation (5) for updating $\hat{z}_j^i[k]$.

Remark 5. *It should be noted that the estimation strategies described previously are actually developed for all the nodes, as one does not know which nodes belong to the set $\mathcal{A}$. The adversarial nodes may or may not choose to follow them.* □

In the next section, we analyze the proposed secure estimation strategy.

IV. Analysis of the Secure Distributed Estimation Strategy

In this section, we provide our main result, which gives a formal proof of asymptotic convergence of the state estimates of the regular nodes to the true state of the plant, under our adopted strategy and under certain conditions on the graph topology. To this end, we first introduce the following definition.

Definition 5. ***(Mode Estimation Directed Acyclic Graph (MEDAG))*** *For each eigenvalue $\lambda_j \in \Lambda_U(\mathbf{M})$, let there exist a spanning subgraph $\mathcal{G}_j = (\mathcal{V}, \mathcal{E}_j)$ of $\mathcal{G}$ with the following properties.*

(i) *If $i \in \{\mathcal{V}\setminus\mathcal{S}_j\} \cap \mathcal{R}$, then $|\mathcal{N}_j^i| \geq 2f+1$, where $\mathcal{N}_j^i = \{l | (l,i) \in \mathcal{E}_j\}$.*[4]
(ii) *There exists a partition of $\mathcal{R}$ into the sets $\{\mathcal{L}_0^j, \mathcal{L}_1^j, \cdots, \mathcal{L}_{T_j}^j\}$, where $\mathcal{L}_0^j = \mathcal{S}_j \cap \mathcal{R}$, and if $i \in \mathcal{L}_m^j$ (where $1 \leq m \leq T_j$), then $\mathcal{N}_j^i \cap \mathcal{R} \subseteq \bigcup_{r=0}^{m-1} \mathcal{L}_r^j$.*

Then, we call $\mathcal{G}_j$ a Mode Estimation Directed Acyclic Graph (MEDAG) for $\lambda_j \in \Lambda_U(\mathbf{M})$. □

If a regular node $i \in \mathcal{L}_m^j$, we say it belongs to level m. The implication of the first property of a MEDAG is that the set $\mathcal{M}_j^i[k]$ (recall that a regular node $i \in \mathcal{V}\setminus\mathcal{S}_j$ only uses estimates from $\mathcal{M}_j^i[k] \subset \mathcal{N}_j^i$ for updating $\hat{z}_j^i[k]$ at time step k) is non-empty $\forall i \in \{\mathcal{V}\setminus\mathcal{S}_j\} \cap \mathcal{R}$. The second property implies that a regular node i in level m (where $1 \leq m \leq T_j$) accepts estimates from only those regular nodes belonging to levels 0 to $m-1$ (and which belong to $\mathcal{M}_j^i[k] \subset \mathcal{N}_j^i$).[5] This gives rise to the acyclic structure of $\mathcal{G}_j$; we later prove that it contains no directed cycles consisting only of regular nodes. The significance of this acyclic structure will be apparent during the convergence analysis.

We now provide a lemma that shall be required for proving our main result.

Lemma 2. *Let Assumption 1 hold. Suppose that the network $\mathcal{G}$ contains a MEDAG $\mathcal{G}_j$ for each $\lambda_j \in \Lambda_U(\mathbf{M})$, and let $\mathcal{N}_j^i$ be the neighbors of node i in $\mathcal{G}_j$. Then, for each regular node $i \in \mathcal{R}$ and each $\lambda_j \in \mathcal{UO}_i$, the LFSE dynamics described by equation (5) ensures that $\lim_{k\to\infty} |\hat{z}_j^i[k] - z_j[k]| = 0$.* □

Proof. As $\mathcal{G}$ contains a MEDAG for each $\lambda_j \in \Lambda_U(\mathbf{M})$, the sets $\{\mathcal{L}_0^j, \mathcal{L}_1^j, \cdots, \mathcal{L}_p^j, \cdots \mathcal{L}_{T_j}^j\}$ form a partition of the set $\mathcal{R}$. We prove by induction on the level number p. For $p = 0$, by definition of the set $\mathcal{L}_0^j$, all the regular nodes in $\mathcal{L}_0^j$ belong to the set $\mathcal{S}_j$, i.e., $\lambda_j \in \mathcal{O}_i$ for each regular node in $\mathcal{L}_0^j$. As Assumption 1 holds true, Lemma 1 holds, and hence each regular node in level 0 can estimate $z_j[k]$ asymptotically. Notice that for any regular node i belonging to a level p, where $1 \leq p \leq T_j$, we have $\lambda_j \in \mathcal{UO}_i$. Consider a regular node i in $\mathcal{L}_1^j$. We partition the set $\mathcal{N}_j^i$ into the sets $\mathcal{U}_j^i[k]$, $\mathcal{L}_j^i[k]$, and $\mathcal{M}_j^i[k]$, such that the sets $\mathcal{U}_j^i[k]$ and $\mathcal{L}_j^i[k]$ contain f nodes each, with the highest and lowest estimate values

[4]Given a regular node $i \in \mathcal{V}\setminus\mathcal{S}_j$, and an undetectable eigenvalue λ_j, the set $\mathcal{N}_j^i$ constructed specifically for λ_j is a certain subset of the original neighborhood of node i in $\mathcal{G}$.

[5]A regular node i also listens to adversarial nodes (if any) in $\mathcal{M}_j^i[k]$.

(for $z_j[k]$) respectively, transmitted to node i at time step k, and $\mathcal{M}_j^i[k]$ contains the rest of the nodes in $\mathcal{N}_j^i$. According to the LFSE dynamics (5), node i only uses estimates from the set $\mathcal{M}_j^i[k]$ to update its own estimate $\hat{z}_j^i[k]$. Notice that based on the properties of a MEDAG, the set $\mathcal{M}_j^i[k]$ is non-empty. Let the error in estimation of $z_j[k]$ for node i be denoted by $e_j^i[k] \triangleq \hat{z}_j^i[k] - z_j[k]$. Subtracting $z_j[k+1]$ from both sides of equation (5), and noting that $z_j[k+1] = \lambda_j z_j[k]$ (based on the decoupled dynamics given by (3)), we obtain

$$\begin{aligned} e_j^i[k+1] &= \lambda_j \sum_{l \in \mathcal{M}_j^i[k]} w_{il}^j[k] \hat{z}_j^l[k] \\ &\quad - \lambda_j \left(\sum_{l \in \mathcal{M}_j^i[k]} w_{il}^j[k] \right) z_j[k] \\ &= \lambda_j \sum_{l \in \mathcal{M}_j^i[k]} w_{il}^j[k] e_j^l[k], \end{aligned} \tag{6}$$

where we used the fact that $\sum_{l \in \mathcal{M}_j^i[k]} w_{il}^j[k] = 1$. Now, consider the following two cases. (i) $\mathcal{M}_j^i[k] \cap \mathcal{A} = \emptyset$, *i.e., there are no adversarial nodes in the set* $\mathcal{M}_j^i[k]$: in this case, all the nodes in the set $\mathcal{M}_j^i[k]$ are regular and belong to $\mathcal{S}_j$ (as $\mathcal{N}_j^i \cap \mathcal{R} \subseteq \mathcal{L}_0^j = \mathcal{S}_j \cap \mathcal{R}$). (ii) $\mathcal{M}_j^i[k] \cap \mathcal{A}$ *is non-empty, i.e., there are some adversarial nodes in the set* $\mathcal{M}_j^i[k]$: based on the f-local adversarial model, it is apparent that each of the sets $\mathcal{U}_j^i[k]$ and $\mathcal{L}_j^i[k]$ contain at least one regular node. Let p and q be two such regular nodes belonging to $\mathcal{U}_j^i[k]$ and $\mathcal{L}_j^i[k]$ respectively. Based on the definitions of the sets $\mathcal{U}_j^i[k]$, $\mathcal{L}_j^i[k]$, and $\mathcal{M}_j^i[k]$, we have $\hat{z}_j^q[k] \leq \hat{z}_j^l[k] \leq \hat{z}_j^p[k]$, and hence $e_j^q[k] \leq e_j^l[k] \leq e_j^p[k]$, for any node $l \in \mathcal{M}_j^i[k]$. Thus, for any $l \in \mathcal{M}_j^i[k]$, we can express $e_j^l[k]$ as a convex combination of the errors $e_j^q[k]$ and $e_j^p[k]$. Analyzing each of the two cases, and referring to equation (6), we infer that at every time-step k, the estimation error $e_j^i[k+1]$ is expressible as a convex combination of the errors of regular nodes in level 0, i.e., regular nodes belonging to the set $\mathcal{S}_j$. Based on Lemma 1, we have that $\lim_{k\to\infty} e_j^l[k] = 0$, $\forall l \in \mathcal{S}_j \cap \mathcal{R}$. Thus, we conclude that $\hat{z}_j^i[k]$ converges asymptotically to $z_j[k]$ for any regular node i in $\mathcal{L}_1^j$. Next, suppose the result holds true for all levels from 0 to p (where $1 \leq p \leq T_j - 1$). It is easy to see that the result holds for all regular nodes in $\mathcal{L}_{p+1}^j$ as well, by noting the following.

- A regular node $i \in \mathcal{L}_{p+1}^j$ has $\mathcal{N}_j^i \cap \mathcal{R} \subseteq \bigcup_{m=0}^p \mathcal{L}_m^j$.
- For each $i \in \mathcal{L}_{p+1}^j$, it holds that every estimate of $z_j[k]$ received (and used for state estimate update) is either from a regular node belonging to any level from 0 to p, or from an adversarial node. In the latter case, based on the LFSE dynamics, this value is sandwiched between the values of two regular nodes (belonging to any level from 0 to p). Since the values of regular nodes in levels $\bigcup_{m=0}^p \mathcal{L}_m^j$ asymptotically converge to $z_j[k]$ (based on our induction hypothesis), the value $\hat{z}_j^i[k]$ will also converge to $z_j[k]$ asymptotically.

Our argument was general and hence holds for any $\lambda_j \in \Lambda_U(\mathbf{M})$. We arrive at the conclusion that any node $i \in \mathcal{R}$ can asymptotically estimate $z_j[k]$ for any eigenvalue $\lambda_j \in \mathcal{UO}_i$. □

We are now in a position to state and prove our main result, which provides sufficient conditions for achieving secure omniscience.

Theorem 1. *Let Assumption 1 hold and suppose that the network $\mathcal{G}$ contains a MEDAG for each $\lambda_j \in \Lambda_U(\mathbf{M})$. Then, the distributed estimation strategy governed by the Luenberger observer based dynamics described by (4), and the LFSE dynamics described by (5), achieves secure omniscience.* □

Proof. Based on Assumption 1, there exists a one-to-one correspondence between the eigenvalues of $\mathbf{M}$, and the components of the transformed state vector $\mathbf{z}[k]$. Accordingly, for each regular node i, the state vector $\mathbf{z}[k]$ can be partitioned into two components $\mathbf{z}_{\mathcal{O}_i}[k]$ and $\mathbf{z}_{\mathcal{UO}_i}[k]$, corresponding to the detectable and undetectable eigenvalues of node i, respectively. As Assumption 1 holds, the result of Lemma 1 holds, and hence $\hat{\mathbf{z}}_{\mathcal{O}_i}[k]$ converges to $\mathbf{z}_{\mathcal{O}_i}[k]$ asymptotically. As Assumption 1 holds and a MEDAG exists for each $\lambda_j \in \mathcal{UO}_i$ (notice that $\mathcal{UO}_i \subseteq \Lambda_U(\mathbf{M})$), the result of Lemma 2 also holds. Consequently, node i can asymptotically estimate each component of the state $\mathbf{z}[k]$ associated with eigenvalues in $\mathcal{UO}_i$, i.e., node i can estimate $\mathbf{z}_{\mathcal{UO}_i}[k]$ asymptotically. Combining these results, we conclude that node i can asymptotically estimate the entire state $\mathbf{z}[k]$, and hence $\mathbf{x}[k]$ also, based on the relation $\mathbf{x}[k] = \mathbf{V}\mathbf{z}[k]$. This completes the proof. □

Having established that secure omniscience can be achieved by each of the regular nodes, we now present a distributed algorithm for constructing a MEDAG for each $\lambda_j \in \Lambda_U(\mathbf{M})$.

V. Distributed MEDAG Construction Algorithm

Recall that the filtering algorithm for secure consensus required a node $i \in \mathcal{V} \backslash \mathcal{S}_j$ to accept estimates from neighbors belonging to the set $\mathcal{M}_j^i[k]$, which was a subset of $\mathcal{N}_j^i$ (the neighbor set of node i in the MEDAG $\mathcal{G}_j$). In this section, we present a distributed algorithm (Algorithm 1) for constructing a MEDAG for each unstable and marginally stable eigenvalue $\lambda_j \in sp(\mathbf{M})$, which in turn outlines the strategy adopted for constructing the set $\mathcal{N}_j^i$ for each $i \in \mathcal{R}$. The construction of these MEDAGs' constitutes the initialization phase of our design, which can then be followed up by the estimation phase described earlier. Algorithm 1 requires every node $i \in \mathcal{R}$ to maintain a counter $c_j(i)$ and a list of indices $\mathcal{N}_j^i$ for each $\lambda_j \in \Lambda_U(\mathbf{M})$. The nodes in $\mathcal{N}_j^i \subseteq \mathcal{N}_i$ will be the parents of node i in the DAG constructed for the estimation of $z_j[k]$. We start with $c_j(i) = 0$ and $\mathcal{N}_j^i = \emptyset$ for each regular node. Each regular source node in $\mathcal{S}_j$ broadcasts a predefined (and arbitrary) message m_j to its outgoing neighbors, sets $c_j(i) = 1$, maintains $\mathcal{N}_j^i = \emptyset$ for all future time, and goes to sleep. Each regular node $i \in \mathcal{V} \backslash \mathcal{S}_j$ waits until it has received m_j from at least $2f+1$ distinct neighbors, at which point it sets $c_j(i) = 1$, appends the labels of each of the neighbors from which it received m_j to $\mathcal{N}_j^i$, broadcasts m_j to its

Algorithm 1 MEDAG Construction Algorithm

The nodes perform a coordinate transformation given by $\mathbf{z}[k] = \mathbf{V}^{-1}\mathbf{x}[k]$.
For each eigenvalue $\lambda_j \in \Lambda_U(\mathbf{M})$ **do**:
Initialization : Initialize $c_j(i) = 0, \quad \mathcal{N}_j^i = \emptyset, \quad \forall i \in \mathcal{R}$. Each node determines whether it belongs to the set $\mathcal{S}_j$.hg
Source nodes transmit : Each regular node in $\mathcal{S}_j$ updates its counter value $c_j(i) = 1$, and transmits a message m_j (e.g. "1") to its outgoing neighbors. Following this step, it does not listen to any other node, i.e., $\mathcal{N}_j^i = \emptyset$ and $c_j(i) = 1, \forall i \in \mathcal{S}_j \cap \mathcal{R}$ for the remainder of the algorithm.
Non-source nodes receive : Each regular node $i \in \mathcal{V} \setminus \mathcal{S}_j$ does the following:

- If $c_j(i) = 0$, *and* node i has received m_j from at least $2f + 1$ distinct neighbors (not necessarily simultaneously) it updates $c_j(i)$ to 1, appends the labels of the neighbors from which it received m_j to $\mathcal{N}_j^i$, and transmits m_j to its outgoing neighbors.
- If $c_j(i) = 1$, it does not change $c_j(i)$, and does not listen to (discards) the information received from its neighbors i.e., it does not update $\mathcal{N}_j^i$.

Return : A set of sets $\{\mathcal{N}_j^i, \lambda_j \in \Lambda_U(\mathbf{M})\}$ for each $i \in \mathcal{R}$.

outgoing neighbors, and goes to sleep. When the algorithm terminates, if we have $c_j(i) = 1$, $\forall i \in \mathcal{R}$, we say that the MEDAG construction algorithm "*terminates*" for λ_j. *The objective of the algorithm is to return the set $\mathcal{N}_j^i$ for every unstable and marginally stable eigenvalue $\lambda_j \in sp(\mathbf{M})$, and $i \in \mathcal{R}$.*

In the next section, we give rigorous graph conditions which guarantee the termination of the MEDAG construction algorithm under arbitrary adversarial behavior. For the following discussion, we characterize the properties of the output of Algorithm 1 if it terminates. Consider the spanning subgraph $\mathcal{G}_j = (\mathcal{V}, \mathcal{E}_j)$ of the original graph $\mathcal{G}$, induced by the sets $\{\mathcal{N}_j^i\}$, $i \in \mathcal{R}$, returned by Algorithm 1. We shall show that $\mathcal{G}_j$ satisfies all the properties of a MEDAG (given in Definition 5).

Proposition 1. *Based on Algorithm 1, the spanning subgraph $\mathcal{G}_j$ of the original graph $\mathcal{G}$ contains no directed cycles where every node belongs to $\mathcal{R}$.* □

Proof. We prove by contradiction. Let there exist a directed cycle $v_i P v_i$, where v_i and the nodes in P belong to $\mathcal{R}$. The path P originates from v_i when node *i* transmits m_j to its outgoing neighbor on path P. Let this occur at time instant $t = k$. Assuming that a node updates its counter value (from 0 to 1) and transmits at the same time instant, node *i* updates its counter value $c_j(i)$ from 0 to 1 and simultaneously transmits m_j at $t = k$. Let the last vertex on path P be v_l. Clearly, node *i* receives information from node *l* at a time instant $t > k$. As a directed edge exists from node *l* to node *i*, it is apparent that node *i* chooses to listen to node *l* even when its counter value $c_j(i)$ is set to 1. This goes against the rules to be followed by a regular node *i* as described by Algorithm 1. Thus, we reach a contradiction. The same argument holds for every node v_i belonging to the set $\mathcal{R} \in \mathcal{G}_j$. □

Proposition 1 justifies the use of the terminology "DAG" regarding the spanning subgraph $\mathcal{G}_j$. Notice that during the construction of a MEDAG for an eigenvalue $\lambda_j \in \Lambda_U(\mathbf{M})$, an adversarial node can misbehave in any of the following ways. (i) It can choose to transmit any message other than the true message m_j. (ii) It can transmit the true message, but out of turn, i.e., before receiving m_j from at least $(2f + 1)$ neighbors. (iii) It can choose not to transmit a message at all. In this context, consider the following result.

Proposition 2. *If Algorithm 1 terminates for $\lambda_j \in \Lambda_U(\mathbf{M})$, then an adversarial node can be detected by a regular node if it violates the rules of Algorithm 1 by sending any message other than m_j to a regular node.* □

Proof. This is a straightforward result which follows directly from the definition of a f-local adversarial model, and by noting that if Algorithm 1 terminates, then $|\mathcal{N}_j^i| \geq 2f + 1$, $\forall i \in \{\mathcal{V} \setminus \mathcal{S}_j\} \cap \mathcal{R}$. □

Thus, Proposition 2 illustrates how the MEDAG construction algorithm can be used to detect any adversarial node which tries to violate the rules of the algorithm, by sending a message other than the true message. However if an adversarial node chooses to send the true message, but does so out of turn, i.e., before receiving the true message from at least $2f + 1$ incoming neighbors (and therefore violating Algorithm 1), then it becomes impossible to detect such an adversarial node based on only local (i.e., neighborhood) knowledge of the graph topology. If a regular node is furnished with the additional information that the network contains N nodes, and that the MEDAG construction algorithm is guaranteed to terminate, then it can detect any adversarial neighbor which chooses not to transmit m_j at all. It can do so by noting that any neighbor which has not transmitted m_j to it within N time-steps (Algorithm 1 is guaranteed to terminate within N time steps under the conditions we provide in the next section) is sure to be an adversarial node.

Interestingly, the proof of asymptotic convergence for Theorem 1 does not require an explicit detection of the adversarial nodes. In other words, the adversarial nodes *can behave arbitrarily* during the MEDAG construction phase, and yet our secure distributed estimation strategy can be shown to succeed under graph conditions which guarantee the termination of the MEDAG construction algorithm. We shall detail such conditions in the next section.

We associate a notion of time with the MEDAG construction algorithm: let a regular node i update its counter value $c_j(i)$ from 0 to 1, and transmit the message m_j at the same time instant $t = k$. Then, we say that node i belongs to level k (for λ_j), denoted by the set $\mathcal{L}_k^j$. We say that node i belongs

to $\mathcal{L}_0^j$ if $i \in \mathcal{S}_j \cap \mathcal{R}$.[6]

Proposition 3. *If Algorithm 1 terminates for $\lambda_j \in \Lambda_U(\mathbf{M})$, the sets $\{\mathcal{L}_0^j, \mathcal{L}_1^j, \cdots, \mathcal{L}_{T_j}^j\}$ form a partition of the set $\mathcal{R}$ in $\mathcal{G}_j$, where T_j denotes the smallest integer such that at time instant T_j, we have $c_j(i) = 1$, $\forall i \in \mathcal{R}$.* □

Proof. Suppose Algorithm 1 terminates for $\lambda_j \in \Lambda_U(\mathbf{M})$. Then, each regular node must update its counter and transmit m_j at some time instant. Thus, it is obvious that $\bigcup_{m=0}^{T_j} \mathcal{L}_m^j = \mathcal{R}$. Also, it is apparent that a regular node cannot update its counter from 0 to 1 and transmit m_j at two distinct time instants (goes against the rules of Algorithm 1). Thus, $\mathcal{L}_m^j \cap \mathcal{L}_n^j = \emptyset$, $\forall m \neq n$. This completes the proof. □

Proposition 4. *If Algorithm 1 terminates for $\lambda_j \in \Lambda_U(\mathbf{M})$, a regular node in $\mathcal{L}_k^j$ (where $1 \leq k \leq T_j$) has at least $f+1$ regular neighbors from the set $\bigcup_{m=0}^{k-1} \mathcal{L}_m^j$ in the spanning subgraph $\mathcal{G}_j$.* □

Proof. Since Algorithm 1 terminates for $\lambda_j \in \Lambda_U(\mathbf{M})$, we have $|\mathcal{N}_j^i| \geq 2f+1$, $\forall i \in \{\mathcal{V} \setminus \mathcal{S}_j\} \cap \mathcal{R}$. Based on the rules of Algorithm 1, if a node $i \in \mathcal{R}$ belongs to $\mathcal{L}_k^j$ (where $1 \leq k \leq T_j$), then $\mathcal{N}_j^i \cap \mathcal{R} \subseteq \bigcup_{m=0}^{k-1} \mathcal{L}_m^j$. Also, since at least $f+1$ of the nodes in the set $\mathcal{N}_j^i$ are regular under the f-local adversarial model, the desired result has to hold. □

Remark 6. *Note that our overall estimation scheme can be broadly decomposed into two phases, namely, the initialization phase and the estimation phase. The initialization phase involves the construction of a MEDAG for each $\lambda_j \in \Lambda_U(\mathbf{M})$, and needs to be implemented just once. Once the initialization phase ends, one can implement the estimation phase, summarized previously in Section III. A merit of the proposed scheme is that each phase of the design admits a fully distributed implementation subject to adversarial behavior.* □

Theorem 2. *If the MEDAG construction algorithm terminates for $\lambda_j \in \Lambda_U(\mathbf{M})$, then there exists a subgraph $\mathcal{G}_j$ satisfying all the properties of a MEDAG.* □

Proof. The result follows trivially from the way $\mathcal{G}_j$ and the sets $\mathcal{L}_m^j$ (for $0 \leq m \leq T_j$) are defined, and from the results of Propositions 1 and 3. □

It follows from the above theorem that if the MEDAG construction algorithm terminates for every $\lambda_j \in \Lambda_U(\mathbf{M})$, then based on Theorem 1, secure omniscience can be achieved by our proposed estimation scheme. In the next section, we provide rigorous conditions on the graph topology which guarantee the termination of the MEDAG construction algorithm.

[6]Note that our strategy allows even some of the source nodes in $\mathcal{S}_j$ to be adversarial.

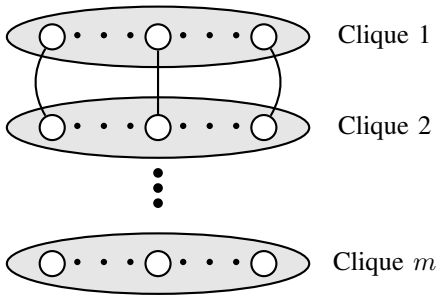

Fig. 1. Illustration of a feasible graph topology. Each clique has size $(3f+1)$. Each node in clique p (where $2 \leq p \leq m-1$) is connected to every node in cliques $p-1$ and $p+1$.

VI. Feasible Graph Topologies

In this section, we characterize a set of feasible graph topologies which guarantee the termination of the MEDAG construction algorithm for each $\lambda_j \in \Lambda_U(\mathbf{M})$. To this end, we borrow the following definition from [9].

Definition 6. *(r-**reachable set**) For a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ and a set $\mathcal{S} \subset \mathcal{V}$, we say $\mathcal{S}$ is an r-reachable set if there exists an $i \in \mathcal{S}$ such that $|\mathcal{N}_i \setminus \mathcal{S}| \geq r$, where $r \in \mathbb{N}_+$.* □

Thus, if a set $\mathcal{S}$ is r-reachable, then it contains a node which has at least r neighbors outside $\mathcal{S}$. We slightly modify the notion of a *strongly-r robust graph* [9] for our case as follows.

Definition 7. *(**strongly** r-**robust graph** w.r.t. $\mathcal{S}_j$) For $r \in \mathbb{N}_+$ and $\lambda_j \in \Lambda_U(\mathbf{M})$, a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is strongly r-robust w.r.t. to the set of source nodes $\mathcal{S}_j$, if for any non-empty subset $\mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{S}_j$, $\mathcal{C}$ is r-reachable.* □

Lemma 3. *The MEDAG construction algorithm terminates for $\lambda_j \in \Lambda_U(\mathbf{M})$ if $\mathcal{G}$ is strongly $(3f+1)$-robust w.r.t. $\mathcal{S}_j$.* □

Proof. We prove by contradiction. Consider any $\lambda_j \in \Lambda_U(\mathbf{M})$ and let $\mathcal{G}$ be strongly $(3f+1)$-robust w.r.t. the set of source nodes $\mathcal{S}_j$. Suppose that the MEDAG construction algorithm for λ_j does not terminate. This implies that there exists a set of *regular* nodes $\mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{S}_j$ which never update their counter $c_j(i)$ from 0 to 1, where $i \in \mathcal{C}$. As $\mathcal{G}$ is strongly $(3f+1)$-robust w.r.t. $\mathcal{S}_j$, it follows that $\mathcal{C}$ is $(3f+1)$-reachable, i.e., there exists a node $i \in \mathcal{C}$ which has at least $3f+1$ neighbors outside $\mathcal{C}$. Under the f-local adversarial model, at most f of these nodes are corrupt, and at least $2f+1$ of them are regular nodes with $c_j(i) = 1$. Thus, at least $2f+1$ regular nodes must have transmitted m_j to node i. Thus, based on the rules of Algorithm 1, node i must have updated $c_j(i)$ from 0 to 1 at some point of time. We thus arrive at a contradiction. □

Based on the above Lemma, we present the main result of this section, which pieces together the previous results presented in this paper, and presents a complete picture of the connection between the feasible graph topologies, and the solution of the secure omniscience achieving problem.

Theorem 3. *If $\mathcal{G}$ is strongly $(3f+1)$-robust w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Lambda_U(\mathbf{M})$, then secure omniscience can be achieved using the proposed estimation strategy.* □

Proof. From Lemma 3, it follows that if $\mathcal{G}$ is strongly $(3f+1)$-robust w.r.t. $\mathcal{S}_j$ for every $\lambda_j \in \Lambda_U(\mathbf{M})$, then the MEDAG construction algorithm terminates for every unstable and marginally stable eigenvalue λ_j. As a result, from Theorem 2, it follows that a MEDAG exists for every $\lambda_j \in \Lambda_U(\mathbf{M})$. Finally, based on the result of Theorem 1, the existence of a MEDAG for every $\lambda_j \in \Lambda_U(\mathbf{M})$ implies that secure omniscience can be achieved using our proposed estimation strategy. This completes the proof. □

Corollary 1. *If $\mathcal{G}$ is strongly $(3f+1)$-robust w.r.t. $\mathcal{S}_j, \forall \lambda_j \in \Lambda_U(\mathbf{M})$, then $|\mathcal{S}_j \cap \mathcal{R}| \geq 2f+1$, i.e., there are at least $(2f+1)$ regular source nodes for every unstable and marginally stable eigenvalue λ_j.* □

As an illustration of a feasible graph topology, consider the network represented by the undirected graph $\mathcal{G}$ shown in Figure 1. $\mathcal{G}$ comprises of m cliques of size $(3f+1)$ each. Further, each node in clique p is connected to every node in cliques $p-1$ and $p+1$ (where $2 \leq p \leq m-1$). For each $\lambda_j \in \Lambda_U(\mathbf{M})$, let the set of source nodes $\mathcal{S}_j$ correspond to one of the m cliques of $\mathcal{G}$. Then, it can be easily verified that $\mathcal{G}$ is strongly $(3f+1)$-robust w.r.t. any $\mathcal{S}_j$, where $\lambda_j \in \Lambda_U(\mathbf{M})$.

A. Connection with Secure Broadcasting Literature

A secure broadcasting algorithm is said to succeed if every regular node in the graph accepts the message transmitted by a single non-adversarial source node. In [9], the authors show that given a designated source node s, the commonly used Certified Propagation Algorithm (CPA) [19] for secure distributed broadcast succeeds under an f-locally bounded adversarial model, if any subset $\mathcal{C} \subseteq \mathcal{V} \setminus \mathcal{S}$ is $(2f+1)$-reachable, where $\mathcal{S} = \{s\} \cup \mathcal{N}_s$ and $\mathcal{N}_s$ denotes the neighborhood of the source node s. Interestingly, in our setting, if the Byzantine attack is limited to the estimation phase only (i.e., all nodes behave regularly during the MEDAG construction phase), then a strongly $(2f+1)$-robust graph w.r.t. $\mathcal{S}_j$ (for each $\lambda_j \in \Lambda_U(\mathbf{M})$) would suffice for achieving secure omniscience.

The above result can be interpreted as follows. For each unstable (or marginally stable) eigenvalue λ_j, the set $\mathcal{S}_j \cap \mathcal{R}$ can be thought of as the source of reliable information for the component $z_j[k]$ of the transformed state vector $\mathbf{z}[k]$. Whereas in broadcast the objective is to transmit a constant message from the source to the rest of the nodes, the objective in distributed estimation is to pass down information about $z_j[k]$ along the edges of the MEDAG constructed for λ_j, from the regular nodes in $\mathcal{S}_j$ to all the regular nodes in $\mathcal{V} \setminus \mathcal{S}_j$. Thus, there exists an analogy between the broadcast problem and the estimation problem. However, unlike distributed broadcast, where the message is a constant, the state $z_j[k]$ is time-varying in the distributed estimation problem, and hence the problem studied in this paper poses a bigger challenge. Thus, it is indeed interesting to note that they can be solved under identical graph conditions (under a specific type of adversarial behavior).

VII. Analysis of the non-adversarial case ($f = 0$)

It is interesting to note that if $f = 0$, i.e., if there are no adversarial nodes in the network, then our proposed estimation strategy succeeds if (i) for every unstable and marginally stable eigenvalue $\lambda_j \in sp(\mathbf{M})$, the set $\mathcal{S}_j$ is non-empty (consider Corollary 1 with $f = 0$); and (ii) every node in $\mathcal{V} \backslash \mathcal{S}_j$ is reachable from at least one source node in $\mathcal{S}_j$. It is easy to see that both these conditions are in fact *necessary* for distributed estimation (for the specific class of LTI systems studied in this work). While the first condition states that for every unstable and marginally stable eigenvalue, there has to exist at least one source node which can detect it, the second condition implies that a node can never estimate the entire state if it does not receive information about an eigenvalue λ_j belonging to its undetectable subspace.

To the best of our knowledge, the literature dealing with the distributed estimation problem (in the absence of adversaries) treats all the nodes in the network identically, in the sense that all the nodes follow the same state-estimate update rule based on consensus dynamics. Our method differs from the existing literature in this area by building on the following key observation: a node needs to rely on consensus for estimating only the portion of the state that corresponds to its undetectable subspace; the rest of the state space can be estimated solely using local measurements. For the class of LTI systems studied in this paper, our approach is appealing because of the following features: (i) it results in a single-time-scale algorithm; (ii) it provides theoretical guarantees regarding the design of asymptotically stable estimators; (iii) it does not require any state augmentation; (iv) it requires only state estimates to be exchanged locally; and (v) it can be implemented in a fully distributed manner.

VIII. Conclusion

We analyzed the problem of distributed state estimation in networks subject to worst-case adversarial attacks for a specific class of LTI systems. We proposed a secure state estimation algorithm and established sufficient conditions for the success of the algorithm. We introduced the notion of a *Mode Estimation Directed Acyclic Graph (MEDAG)*, and showed that it plays a key role in guaranteeing asymptotic convergence. Accordingly, we presented a distributed algorithm for constructing a MEDAG for each unstable and marginally stable eigenvalue of the plant. Next, we characterized a set of feasible graph topologies which guaranteed the termination of the MEDAG construction algorithm, and in turn, the success of our overall secure estimation scheme. Finally, we discussed that for a non-adversarial setting, our method leads to the design of distributed observers possessing several attractive features simultaneously.

Future work in this area would aim towards the extension of our existing method to systems with more general plant dynamics. In this work, we provided sufficient conditions on the graph topology which guarantee the success of

our method. However, the problem of obtaining a single necessary and sufficient condition under the most general plant dynamics for solving the secure distributed estimation problem remains open.

REFERENCES

[1] R. Olfati-Saber, "Distributed Kalman filter with embedded consensus filters," in *Proceedings of the 44th IEEE Conference on Decision and Control and European Control Conference*, 2005, pp. 8179–8184.

[2] ——, "Distributed Kalman filtering for sensor networks," in *Proceedings of the 46th IEEE Conference on Decision and Control*, 2007, pp. 5492–5498.

[3] U. A. Khan and A. Jadbabaie, "On the stability and optimality of distributed Kalman filters with finite-time data fusion," in *Proceedings of the American Control Conference*, 2011, pp. 3405–3410.

[4] U. Khan, S. Kar, A. Jadbabaie, and J. M. Moura, "On connectivity, observability, and stability in distributed estimation," in *Proceedings of the 49th IEEE Conference on Decision and Control*, 2010, pp. 6639–6644.

[5] U. A. Khan and A. Jadbabaie, "Collaborative scalar-gain estimators for potentially unstable social dynamics with limited communication," *Automatica*, vol. 50, no. 7, pp. 1909–1914, 2014.

[6] S. Park and N. C. Martins, "Necessary and sufficient conditions for the stabilizability of a class of LTI distributed observers," in *Proceedings of the 47th IEEE Conference on Decision and Control*, 2012, pp. 7431–7436.

[7] ——, "A class of lti distributed observers for LTI plants: Necessary and sufficient conditions for stabilizability," *arXiv preprint arXiv:1401.0926*, 2014.

[8] B. D. Anderson and D. J. Clements, "Algebraic characterization of fixed modes in decentralized control," *Automatica*, vol. 17, no. 5, pp. 703–712, 1981.

[9] H. Zhang and S. Sundaram, "Robustness of information diffusion algorithms to locally bounded adversaries," in *Proceedings of the American Control Conference*, 2012, pp. 5855–5861.

[10] H. J. LeBlanc, H. Zhang, X. Koutsoukos, and S. Sundaram, "Resilient asymptotic consensus in robust networks," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 4, pp. 766–781, 2013.

[11] L. Tseng, N. Vaidya, and V. Bhandari, "Broadcast using certified propagation algorithm in presence of byzantine faults," *arXiv preprint arXiv:1209.4620*, 2012.

[12] S. Sundaram and B. Gharesifard, "Consensus-based distributed optimization with malicious nodes."

[13] L. Su and N. Vaidya, "Byzantine multi-agent optimization: Part i," *arXiv preprint arXiv:1506.04681*, 2015.

[14] I. Shames, A. M. Teixeira, H. Sandberg, and K. H. Johansson, "Distributed fault detection for interconnected second-order systems," *Automatica*, vol. 47, no. 12, pp. 2757–2764, 2011.

[15] I. Matei, J. S. Baras, and V. Srinivasan, "Trust-based multi-agent filtering for increased smart grid security," in *Proceedings of the 20th Mediterranean Conference on Control & Automation*, 2012, pp. 716–721.

[16] T. Jiang, I. Matei, and J. Baras, "A trust based distributed kalman filtering approach for mode estimation in power systems," in *Proceedings of the First Workshop on Secure Control Systems*, 2010.

[17] U. Khan and A. M. Stankovic, "Secure distributed estimation in cyber-physical systems," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 2013, pp. 5209–5213.

[18] D. Dolev, N. A. Lynch, S. S. Pinter, E. W. Stark, and W. E. Weihl, "Reaching approximate agreement in the presence of faults," *Journal of the ACM (JACM)*, vol. 33, no. 3, pp. 499–516, 1986.

[19] C.-Y. Koo, "Broadcast in radio networks tolerating byzantine adversarial behavior," in *Proceedings of the twenty-third annual ACM symposium on Principles of distributed computing*. ACM, 2004, pp. 275–282.

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

[21] C.-T. Chen, *Linear System Theory and Design*. Oxford University Press, Inc., 1995.

APPENDIX

Vulnerability to Adversarial Attacks

In this section, we study the performance of an existing *'trust-based'* secure distributed estimation scheme in the face of adversarial attacks. Specifically, we present an example which demonstrates how a carefully crafted attack can cause an estimation scheme relying on the notion of 'belief divergence' [16] to fail (similar trust-based schemes have been used in [15], [17]). The example shall also illustrate how the method proposed in this paper succeeds under such an attack.

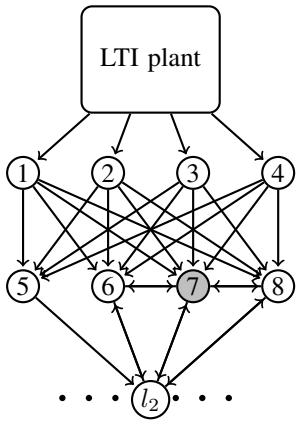

Fig. 2. Network topology for illustrating the attack. Node 7 is the only Byzantine adversary in the network.

a) Updating Trust Values based on Belief Divergence: We briefly discuss the mechanism for updating trust values based on the notion of belief divergence, as employed in [16]. First, based on the state estimates received from its neighbors, each node $i \in \mathcal{V}$ computes a belief divergence d_{ij}, for each node j in its neighborhood, according to the formula

$$d_{ij} = \frac{1}{|\mathcal{N}_i| - 1} \sum_{k \in \mathcal{N}_i} ||\hat{\mathbf{x}}_j - \hat{\mathbf{x}}_k||^2. \tag{7}$$

Next, the trust values are updated by node i using the following formula:

$$T_{ij} = c_i - d_{ij},\ j \in \mathcal{N}_i, \tag{8}$$

where c_i is a positive constant chosen as

$$c_i > max\{d_{ij} \mid j \in \mathcal{N}_i\}. \tag{9}$$

Such a choice of c_i ensures that the trust values in equation (8) are always positive. The normalized versions of the trust values T_{ij} are computed according to the formula

$$p_{ij} = \frac{T_{ij}}{\sum_{k \in \mathcal{N}_i} T_{ik}}. \tag{10}$$

Noting from the above formula that the normalized trust-values are not necessarily zero for nodes with large belief divergence, a thresholding scheme is introduced on the normalized trust values. Accordingly, if $p_{ij} < p_i^{min}$, then the trust value T_{ij} is set to zero. Here, p_i^{min} represents

the minimum acceptable value for p_{ij} and its lower bound is chosen to be inversely proportional to $|\mathcal{N}_i|$. Finally, the weights used in the consensus step are updated using the following equation

$$w_{ij} = \frac{T_{ij}}{\sum_{k\in\mathcal{N}_i} T_{ik}}. \tag{11}$$

In [16], the authors provide a distributed Kalman filtering algorithm which essentially involves a local update rule of the form

$$\boldsymbol{\phi}_i = \boldsymbol{\zeta}_i + \mathbf{L}_i(\mathbf{y}_i - \mathbf{C}_i\boldsymbol{\zeta}_i), \tag{12}$$

where $\boldsymbol{\phi}_i$ is an intermediate Kalman estimate of the true state and $\mathbf{L}_i$ is a time-varying gain matrix which needs to be updated (for more details refer to [16]). Based on the belief divergence scheme described previously, the state is updated using a consensus rule given by

$$\hat{\mathbf{x}}_i = \sum_{j\in\mathcal{N}_i} w_{ij}\boldsymbol{\phi}_j, \tag{13}$$

followed up by the update $\boldsymbol{\zeta}_i = \mathbf{A}\hat{\mathbf{x}}_i$.

b) Attacking the Trust-based Algorithm: Consider a scalar plant with dynamics $x[k+1] = 1.5x[k]$, and network topology $\mathcal{G}$ given by Figure 2. The nodes 1,2,3 and 4 have identical measurements $y_i[k] = x[k], i = 1, 2, 3, 4$, while the rest of the nodes have no measurements. Let $\mathcal{S} = \{1, 2, 3, 4\}$. Node 7 is the only Byzantine adversary in the network. There are M identical nodes l_2 in $\mathcal{G}$. We shall show that although the graph $\mathcal{G}$ has a majority of regular nodes in the neighborhood of any node, the single Byzantine adversary (node 7) can cause the estimates of nodes 6,8 and each of the M identical nodes l_2, to diverge from the true state of the plant, under the 'trust-based' weight assignment scheme based on the notion of belief divergence. Since we consider a deterministic setting, equation (12) amounts to running a standard Luenberger observer with a fixed gain. Accordingly, the nodes in $\mathcal{S}$ run the following observer for estimating and predicting the state:

$$\hat{x}_i[k+1] = 1.5\hat{x}_i[k] + (y[k] - \hat{x}_i[k]), i \in \mathcal{S}, \tag{14}$$

where $\hat{x}_i[k]$ denotes the estimate of $x[k]$ maintained by node i at time-step k. It can be easily verified that the resulting estimation error dynamics is asymptotically stable. Each regular node i in $\mathcal{V}\backslash\mathcal{S}$ has no measurements and hence relies purely on consensus; based on equation (13), and noting that $A = 1.5$ in this example, the consensus equation is simply $\hat{x}_i[k+1] = 1.5\sum_{j\in\mathcal{N}_i} w_{ij}\hat{x}_j[k]$, where the weights w_{ij} are decided based on the notion of belief divergence at every time instant. If $\mathcal{G}$ contains no adversarial nodes, this scheme shall always achieve omniscience for any choice of M. Next, we outline the strategy to be followed by the adversarial node 7.

The Byzantine adversary does the following at each time-step k: (i) it transmits $\hat{x}_{l_2}[k] + \epsilon_6$ to nodes 6 and 8; and (ii) it transmits $\hat{x}_6[k] + \epsilon_{l_2}$ to each of the M identical nodes l_2. Here, ϵ_6 and ϵ_{l_2} are small deviations chosen appropriately by node 7 such that the estimates transmitted by it are used in the consensus update by nodes 6,8 and the M identical nodes l_2 (i.e, these values are chosen in a way such that node 7 gets assigned a *non-zero trust* by its neighbors).[7] Note that node 7 can transmit different estimates to its neighbors at the same time instant since it is assumed to be a Byzantine adversary. The insight behind this attack strategy is as follows: with respect to each of the nodes 6 and 8, node 7 pretends to behave like one of the regular nodes l_2. Similarly, with respect to each of the identical nodes l_2, node 7 pretends

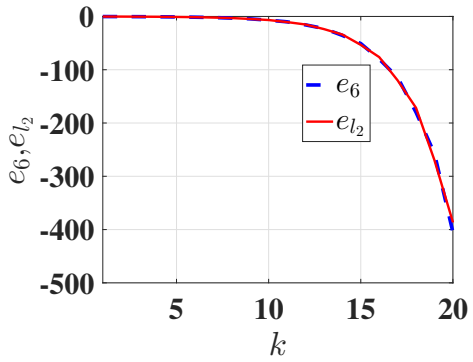

Fig. 3. Estimation errors of node 6 and the M identical nodes l_2 when $M = 4$.

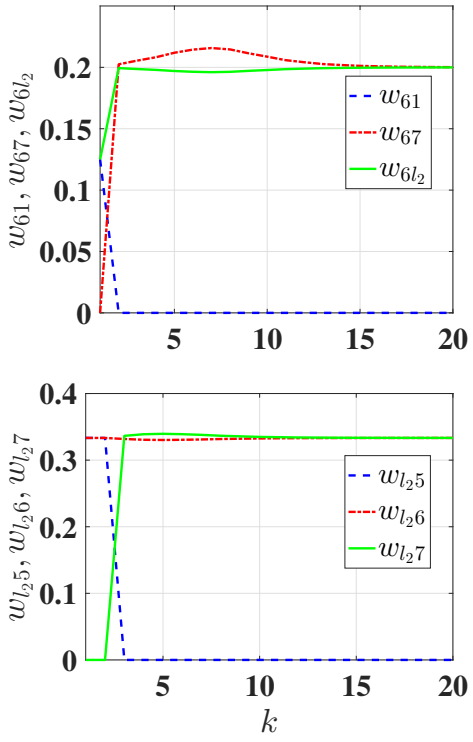

Fig. 4. Consensus weights when $M = 4$. (*Top*) Weights assigned by node 6. (*Bottom*) Weights assigned by the M identical nodes l_2.

[7]Note that since node 7 transmits the same value to nodes 6 and 8 at every time-step, the estimates of nodes 6 and 8 evolve identically (based on the graph topology). Similarly, the estimates of each of the M nodes l_2 evolve identically.

to behave like node 6 or node 8. The ultimate objective is to drive out the nodes $\mathcal{S}$ from the set of trusted nodes in the neighborhoods of nodes 6 and 8, and drive out node 5 (the estimate of node 5 is not corrupted by node 7 and is hence sure to converge) from the trusted neighborhood of each of the identical nodes l_2. By doing so, and by corrupting the estimates of the nodes 6,8 and the set of identical nodes l_2, the Byzantine adversary causes their estimates to diverge.

Consider the case when $M = |\mathcal{S}| = 4$. For this case, our experiments reveal that the adversary succeeds in making the estimates of nodes 6 and l_2 diverge. This is achieved as follows. By corrupting the estimates of the 4 identical nodes l_2 in the neighborhood of node 6 (and node 8), the adversary causes these estimates to be aligned with its own estimate and to slowly drift away from the estimates of the 4 nodes in $\mathcal{S}$ (the estimates of the nodes in $\mathcal{S}$ evolve identically). As this process continues, the estimates of a majority of the nodes in the neighborhoods of nodes 6 and 8 become corrupted. Consequently, based on the notion of belief divergence, nodes 6 and 8 assign gradually decreasing trust values (and hence consensus weights) to the nodes in $\mathcal{S}$, eventually causing these weights to drop to zero. Figures 3 and 4 illustrate this phenomenon. Based on the evolution of weights in Figure 4, notice that the nodes in $\mathcal{S}$ (which are sure to converge) are assigned zero weights at steady state by node 6 (as w_{61} settles to zero). Similarly, node 5 (which is not corrupted by the adversary) gets assigned a zero weight by the nodes l_2 (as $w_{l_2 5}$ settles to zero). Node 6 treats node 7 identically as the nodes l_2, and hence each of these nodes get assigned a weight of $\frac{1}{(M+1)}$ (which in this case equals 0.2 as $M = 4$). Similarly, nodes l_2 treat node 7 identically as the nodes 6 and 8, and hence these nodes get assigned a weight of $\frac{1}{3}$ each. This example thus illustrates how a smart adversary can exploit the graph topology and breach a 'trust-based' security scheme, even for networks with high-connectivity such as the one considered in this example. An identical trend is noted for $M > |\mathcal{S}|$.

For each of the simulations, we have used the following parameters: $p_6^{min} = \frac{1}{|\mathcal{N}_6|}, p_{l_2}^{min} = \frac{1}{|\mathcal{N}_{l_2}|}$, c_6 and c_{l_2} in equation (9) are chosen to be 0.5 higher than the maximum belief divergences in their neighborhood, and ϵ_6 and ϵ_{l_2} (the small deviations injected by the adversary) are each set to 0.05. The true state $x[k]$ has an initial value of 0.2, and the initial estimates maintained by each of the source nodes is zero.

Notice that each of the nodes in $\mathcal{S}$ can detect the unstable eigenvalue $\lambda = 1.5$, and hence they act as the source nodes for $\lambda = 1.5$. It is easy to see that the network in Figure 2 is *strongly* $(3f + 1)$*-robust* with respect to the source nodes $\mathcal{S}$ (here f = 1). Hence, irrespective of any strategy adopted by the single adversary (node 7), the secure distributed estimation scheme presented in this paper shall allow each regular node to achieve omniscience based on Theorem 3.