2021 60th IEEE Conference on Decision and Control (CDC)
December 13-15, 2021. Austin, Texas

Decentralized Resilient State-Tracking

Yanwen Mao Paulo Tabuada

2021 60th IEEE Conference on Decision and Control (CDC) | 978-1-6654-3659-5/21/$31.00 ©2021 IEEE | DOI: 10.1109/CDC45484.2021.9683313

***Abstract*— We study the decentralized resilient state-tracking problem in which each node in a network has the objective of tracking the state of a linear dynamical system based on its local measurements and information exchanged with its neighboring nodes, despite an attack on some of the nodes. We propose a novel algorithm that solves the decentralized resilient state-tracking problem by relating it to the dynamic average consensus problem. Compared with existing solutions in the literature, our algorithm provides a solution for the most general class of decentralized resilient state-tracking problem instances.**

I. Introduction

The problem studied in this paper, the Decentralized Resilient State-Tracking (DRST) problem, asks for the possibility of tracking the state of a linear dynamical system monitored by a network of sensors, despite an attack altering the measurements of some sensors. The DRST problem is motivated by the several attacks on cyber-physical systems that have been reported in the past decade [1], [2]. Different from the closely-related and well-studied Resilient State-Reconstruction (RSR) problem [3]–[6], which aims at securely reconstructing the state of a linear dynamical system at a centralized location which has access to all the measurements, the DRST problem is more challenging since each node relies solely on information exchanged with its neighboring nodes and obtained through its sensors.

The solvability of the RSR problem was settled in [3], [5], [7]. In particular, the notion of sparse observability defined in [5] for discrete-time systems and a similar notion for continuous-time systems in [7], explicitly reveals the connection between the solvability of an RSR problem instance and the redundancy of sensing resources. Although a class of RSR problems admits a polynomial-time solution [4], [8], [9], the general RSR problem is intrinsically an NP-hard problem.

Compared with the RSR problem, the DRST problem is much more complex and the current understanding of this problem is limited. To the best of the authors' knowledge, only three papers addressed the DRST problem. We can find in [10] a formalization of the DRST problem as a distributed convex optimization problem with time-varying objective function which leads to a high-gain observer that tracks the state with the help of the "blended dynamics approach" introduced in [11]. However, the results stated in [10] require a special property of the system dynamics, named Scalar Decomposability (SD). Intuitively, SD enables one to decompose a DRST problem into multiple subproblems each associated with a scalar system. The DRST problem was first addressed in [8], in which a local filter was proposed to force the estimate of an attack-free sensor to lie in the convex hull of the estimates of its attack-free neighbors. Although this strategy allows extensions to defend against, not only attacks on measurements but also attacks on nodes, it only offers a solution for the case where the number of attacked sensors is not large in the neighborhood of any node. Moreover, the result in [8] also relies on an assumption very similar to SD, but less restrictive. Lastly, paper [12] offers an alternative solution of the DRST problem under some restrictions on the system dynamics and the network topology. Compared with [8] and [10], the state observer proposed in [12] does not need the SD property but requires a much higher communication rate than the sampling rate, which could be demanding in real applications.

In this paper, we solve the DRST problem by relating it to the dynamic average consensus problem. A thorough literature review of the dynamic average consensus problem is provided in the tutorial paper [13] and we refer interested readers to that paper. Our solution is based on the simple observation that it is not necessary for nodes to have access to all the measurements in the network to track the state. Instead, it suffices for the nodes to have access to a suitably compressed version of all the measurements. This means that if each sensor has access to the compressed measurements, then the DRST problem can be solved by having each sensor execute a slight modification of any existing algorithm for the RSR problem that does not require assumptions beyond those listed in Section III. The key step of our solution, which is making the compressed measurements accessible to all sensors, is achieved by invoking results in the dynamic average consensus literature. In particular, we adopt and extend the solution in [14] to force all nodes to reach consensus on the compressed measurements, as we will discuss in Section VI.

II. Preliminaries and Notations

In this section we introduce the notions used throughout the paper.

A. Basic Notions

We denote by $|S|$ the cardinality of a set S. For any two sets S and S', the set subtraction $S \backslash S'$ is the set defined by $S \backslash S' = \{s \in S | s \notin S'\}$.

This work was funded in part by the Army Research Laboratory under Cooperative Agreement W911NF-17-2-0196 and in part by NSF grant 1705135.

Yanwen Mao and Paulo Tabuada are with the Department of Electrical and Computer Engineering, University of California, Los Angeles, California, 90095, USA {yanwen.mao,tabuada}@ucla.edu

Let $\mathbb{R}$, $\mathbb{N}$, and $\mathbb{C}$ denote the sets of real, natural, and complex numbers, respectively. The support of $v \in \mathbb{R}^p$, denoted by $\text{supp}(v)$, is the set of indices of the non-zero entries of v, i.e., $\text{supp}(v) = \{i \in \{1, 2, \dots, p\} | v_i \neq 0\}$. For a scalar $s \in \mathbb{N}$ we say v is $s-$sparse if $|\text{supp}(v)| \leq s$. Also, we define the all-ones vector $\mathbf{1}_n = (1, 1, \dots, 1)^T$ and I_n to be the identity matrix of order n.

Lastly, for an infinitely differentiable function $y(t)$ of time, we denote by $y^{(n)}(t)$ the n-th derivative of $y(t)$, for any positive $n \in \mathbb{N}$.

B. Matrix Related Notions

For a real square matrix A, we denote by $\lambda_{\max}(A)$ and $\lambda_{\min}(A)$ the eigenvalue with the largest and smallest magnitude, respectively. We denote by $A \otimes B$ the Kronecker product of two real matrices A and B. We will also refer to matrices where only the number of rows or columns is specified using the notation $A \in \mathbb{R}^{m \times *}$ or $A \in \mathbb{R}^{* \times n}$.

Consider a set Q of indices and a matrix K, the matrix K_Q is obtained by removing any row in K that is not indexed by Q.

C. Graph Related Notions

Here we review some of the basic notions of graph theory. A weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ is a triple consisting of a set of vertices $\mathcal{V} = \{v_1, v_2, \dots, v_p\}$ with cardinality p, a set of edges $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$, and a weighted adjacency matrix $\mathbf{A} \in \mathbb{R}^{p \times p}$ which we will define in the coming paragraph. The set of neighbors of a vertex $i \in \mathcal{V}$, denoted by $\mathcal{N}_i = \{j \in \mathcal{V} | (i, j) \in \mathcal{E}\}$ is the set of vertices that is connected to i by an edge. To clarify, we assume each vertex is not a neighbor of itself, i.e., $(i, i) \notin \mathcal{E}$ for any i.

The weighted adjacency matrix $\mathbf{A}$ of the graph $\mathcal{G}$ is defined entry-wise. The entry in the i-th row and j-th column, a_{ij}, satisfies $a_{ij} > 0$ if $(i, j) \in \mathcal{E}$ and otherwise $a_{ij} = 0$. Since the graph is undirected, $a_{ij} = a_{ji}$ for any i, j ranging from 1 to p which which results in $\mathbf{A}$ being a symmetric matrix. The degree matrix $\mathbf{D} \in \mathbb{R}^{p \times p}$ of the graph $\mathcal{G}$ is a diagonal matrix with its i-th diagonal element defined by $d_{ii} = \sum_{j=1}^{p} a_{ij}$. The Laplacian matrix $\mathcal{L}$ of the graph $\mathcal{G}$ is defined by $\mathcal{L} = \mathbf{D} - \mathbf{A}$, which is known to be symmetric if the graph is undirected, positive semi-definite and having $\text{span}\{\mathbf{1}_p\}$ as its kernel.

III. Problem Formulation and Key Idea

In this section we introduce the decentralized secure state-reconstruction problem.

A. System Model

We consider a linear time-invariant system monitored by a network of p nodes which are subject to attacks:

$$\begin{aligned} \dot{x} &= Ax, \\ y_i &= C_i x + e_i, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^n$ is the system state, $y_i \in \mathbb{R}$ is the measurement of node i, which is assumed to be a scalar, where $i \in P \triangleq \{1, 2, \dots, p\}$, and the matrices A and C_i have appropriate dimensions. Note that in the most general case sensors have vector measurements (i.e., y_is are vectors). In the context of this paper we adopt the scalar measurement assumption for the sake of simplicity. The scalar $e_i \in \mathbb{R}$ models the attack on node i. If the node i is attacked by an adversary then e_i is arbitrary, otherwise, e_i remains constantly equal to zero, and $y_i = C_i x$ which means node i has the correct measurements. We also assume that the adversary is omniscient, i.e., it knows the system state x and the measurements y_i from all nodes. The only assumption we make about the adversary is that it can only attack a fixed set of at most s nodes. This attacked set is unknown to any attack-free node. All of the above signals x, y_i, and e_i are time-varying, but for the sake of simplicity we drop the time index.

Furthermore, by collecting the measurement together with its $(n-1)$-th derivatives[1] from each sensor we can write the output of the system in the more compact form:

$$Y_i = \mathcal{O}_i x + E_i, \quad i = 1, \dots, p, \tag{2}$$

where $Y_i = \begin{bmatrix} y_i & \dots & y_i^{(n-1)} \end{bmatrix}^T$, $E_i = \begin{bmatrix} e_i & \dots & e_i^{(n-1)} \end{bmatrix}^T$, and $\mathcal{O}_i = \begin{bmatrix} C_i^T & (C_i A)^T & \dots & (C_i A^{n-1})^T \end{bmatrix}^T$ is the observability matrix of node i. This motivates us to write down the following stacked-form expression of the system (1):

$$\begin{aligned} \dot{x} &= Ax, \\ Y &= \mathcal{O} x + E, \end{aligned} \tag{3}$$

where Y, $\mathcal{O}$, and E are obtained by stacking vertically each Y_i, $\mathcal{O}_i$, and E_i, respectively, for $i \in \{1, 2, \dots, p\}$, i.e.:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_p \end{bmatrix} \in \mathbb{R}^{pn}, \; \mathcal{O} = \begin{bmatrix} \mathcal{O}_1 \\ \mathcal{O}_2 \\ \vdots \\ \mathcal{O}_p \end{bmatrix} \in \mathbb{R}^{pn \times n}, \; E = \begin{bmatrix} E_1 \\ E_2 \\ \vdots \\ E_p \end{bmatrix} \in \mathbb{R}^{pn}.$$

We also note that since the adversary can only attack at most s nodes, at most s blocks in E are non-zero. Furthermore, we define $C = \begin{bmatrix} C_1^T & C_2^T & \dots & C_p^T \end{bmatrix}^T$ for future use.

We assume that the communication between nodes in the network can be modeled by an undirected graph. Each node is modeled by a vertex $i \in \mathcal{V}$, and a communication link from node i to node j is modeled by an edge $(i, j) \in \mathcal{E}$ from vertex i to j. Since we assume the graph is undirected, $(i, j) \in \mathcal{E}$ implies $(j, i) \in \mathcal{E}$ which shows that node j can also send messages to node i.

B. Assumptions

Here we list all the assumptions we use in this paper. Some of them have already been discussed when introducing the adversary model. Generally speaking we classify all the assumptions into two categories. Assumptions 1-4 are widely

[1]Since the output of an attack-free node is infinitely differentiable, we assume, without loss of generality, that e_i is also infinitely differentiable since, otherwise, the attacked nodes could be identified by simply checking the differentiability properties of the output. The local sanity check which we will introduce in Section V has the same spirit.

We note that, although these derivatives cannot be computed in practice, all the result in this paper admit a discrete-time version, thus making our approach practical.

accepted in the literature and we also use them as part of our problem setting. Assumptions 5 and 6 are adopted to ease the analysis but are unnecessary.

Assumption 1. *The adversary is only able to change the measurements of the attacked nodes. Each attacked node still executes its algorithm correctly.*

Assumption 2. *The adversary is only able to attack at most s nodes. The set of attacked nodes remains constant over time. However, the identity of the attacked nodes is unknown to any node in the network.*

Assumption 3. *The system dynamics are known to all nodes in the network.*

Assumption 4. *The network can be modeled by a communication graph which is time-invariant, undirected and connected.*

The assumptions 1, 2, 3, and 4 are in line with the assumptions in [10] and [12] except that we also assume each node knows the C_i matrices of all other nodes throughout the network. To simplify our analysis we also introduce the following assumptions:

Assumption 5. *All the measurements y_i are scalars.*

Assumption 6. *The real parts of all eigenvalues of A are positive.*

Note that assumptions 5 and 6 are not necessary and we leave for future work how to suitably modify the results so as not to rely on them.

C. The Decentralized Resilient State-Tracking Problem

In this section we provide the definition of the DRST problem.

In plain words, to solve the DRST problem, each node i must maintain a state estimate $\hat{x}_i(t)$ which converges asymptotically to the true state $x(t)$. We also refer to this property by saying that $\hat{x}_i(t)$ tracks $x(t)$. The rigorous definition of the DRST problem is as follows:

Definition 1 (Decentralized Resilient State-Tracking Problem)**.** *Consider a linear system subject to attacks* (1) *satisfying assumptions 1-3 and 5-6, and a communication network satisfying assumption 4. The decentralized resilient state-tracking problem asks for an algorithm running at each node i with measurements y_i and messages from neighboring nodes as its input, such that its output $\hat{x}_i(t)$ satisfies:*

$$\lim_{t\to\infty} \|\hat{x}_i(t) - x(t)\| = 0.$$

□

D. Key Idea

The key idea for solving the DRST problem is based on the simple observation that instead of having access to measurements $Y = \begin{bmatrix} Y_1^T & Y_2^T & \dots & Y_p^T \end{bmatrix}^T$ of all sensors, a compressed version $(D \otimes I_n)Y$ of the measurements may suffice to reconstruct the state, where the compression matrix $D \in \mathbb{R}^{v\times p}$ reduces the measurements from $\mathbb{R}^p$ to $\mathbb{R}^v$ with $v \leq p$. Compression is possible, in most cases, and thus v will be strictly smaller than p. We will elaborate on the feasible choices for a compression matrix D in Section IV.

Equipped with the observation that the compressed version of measurements $(D \otimes I_n)Y$ suffices to reconstruct state, it is natural to ask: how can each node have access to the compressed measurements? We show how to reformulate this problem as a dynamic average consensus problem in Section V and an algorithm for each node to track $(D\otimes I_n)Y$ is provided in Section VI.

Lastly, to reconstruct the state x from the compressed measurements $(D \otimes I_n)Y$ at each node, we may employ any suitably modified algorithm for the RSR problem that does not require additional assumptions. The modification is required since compression slightly changes the attack model as the effect of the attack is altered by the compression matrix. After performing such modifications, it follows that, as each sensor's estimate of the compressed measurements converges, so does the state reconstructed from the estimated compressed measurements. However, due to the space limitations, we can only provide our key results at the end of Section VI without offering any details. These three steps provide a solution to the DRST problem.

IV. Design of the Compression Matrix

There are two considerations involved in the choice of the compression matrix D. On the one hand, in order to reduce communications and storage, we want the D matrix to have the least possible number of rows. On the other hand, the compressed measurements $(D\otimes I_n)Y$ must provide enough information for each node to correctly reconstruct the state. We start with the definition of detectability and sparse detectability with respect to a matrix, which is a generalization of sparse observability [5], [7] as well as sparse detectability [15] but stronger, as we will very soon see.

Definition 2 (Detectability [16])**.** *A pair (A, C) is detectable if all the unobservable eigenvalues of A are stable.* □

Definition 3 (Sparse detectability with respect to a matrix)**.** *Consider the system* (1)*, a matrix $D \in \mathbb{R}^{v\times p}$, and define the following set:*

$$\mathbf{P}_s = \{L \in \mathbb{R}^{*\times v} | \ker(L) = D(\text{span } V), V \subseteq \mathbf{E}_p, |V| \leq s\}.$$

The sparse detectability index of the system (1) *with respect to D is the largest integer k such that the pair (A, LDC) is detectable for any $L \in \mathbf{P}_k$. When the sparse detectability index with respect to D is k, we say that system* (1) *is k-sparse detectable with respect to D.* □

Also, if a pair (A, C) is $s-$sparse detectable with respect to I, then we say that the pair (A, C) is $s-$sparse detectable.

Lemma 1. *Any non-$s-$sparse detectable pair (A, C) is not $s-$sparse detectable with respect to any matrix of appropriate dimension.* □

Lemma 2. *Consider the linear system subject to attacks defined in* (1) *satisfying assumptions 1-4, the compressed measurement* $(D \otimes I_n)Y$ *suffices to track the state* x*, if and only if* (A, C) *is* $2s-$*sparse detectable with respect to* D*.* □

The proofs of *all* results in this document can be found in [17].

Corollary 1. *The measurements* $Y_1, \ldots, Y_p$ *suffice to track the state* x *if and only if* (A, C) *is* $2s-$*sparse detectable.* □

This corollary is obtained by noting that if the pair (A, C) is $2s-$sparse detectable, then we can always find a compression matrix D such that the pair (A, C) is $2s-$sparse detectable with respect to D, for example, by taking D to be the identity matrix. On the other hand, in the light of Lemma 1, if the pair (A, C) is not $2s-$sparse detectable, then the non-existence of such a matrix D implies the unsolvability of the DRST problem.

V. Reduction to Dynamic Average Consensus

One key step of our solution to the DRST problem is to have each node tracking the compressed measurements $(D \otimes I_n)Y$. To do this, we ask each node to maintain an estimate vector $J = \begin{bmatrix}(J^1)^T & (J^2)^T & \ldots & (J^v)^T\end{bmatrix}^T \in \mathbb{R}^{vn}$, whose j-th block, J^j, tracks the j-th linear combination of measurements $\sum_i d_{ji}Y_i$, where d_{ji} is the entry at the j-th row and i-th column of D.

In this section and in the one that follows, we focus on the problem of tracking $(D_1 \otimes I_n)Y$, where D_1 is the first row of D, which could also be written as $\sum_i d_{1i}Y_i$. For technical reasons we will actually be tracking $\frac{1}{p}\sum_i d_{1i}Y_i$, which serves the same purpose since by assumption 3 the value of p is known to all nodes. Note that any algorithm that can track $\frac{1}{p}\sum_i d_{1i}Y_i$ can be extended to track $\frac{1}{p}(D \otimes I_n)Y$ but running v concurrent copies, each with a different set of weights $\{d_{ji}\}$. We observe that this problem can be seen as an instance of the dynamic average consensus problem [13]. In brief, suppose that each agent in the network has a local reference signal $\phi_i : [0, \infty) \to \mathbb{R}^n$. The dynamic average consensus algorithm asks for an algorithm that allows individual agents to track the time-varying average of the reference signals, given by:

$$u^{\text{avg}}(t) = \frac{1}{p}\sum_{i=1}^{p} \phi_i(t). \tag{4}$$

In our problem setting, we let $\phi_i(t) = d_{1i}Y_i(t)$ and what we want to track is $\frac{1}{p}\sum_{i=1}^{p}\phi_i(t)$. In other words, we may adopt any algorithm that solves the dynamic average consensus algorithm which thereby enabling all nodes to track $(D \otimes I_n)Y$.

However, in the setting of dynamic average consensus problem, no knowledge about reference signals ϕ_i is assumed, whereas in our problem, for an attack-free node i, we have the following:

$$\dot{Y}_i = \underbrace{\begin{bmatrix} 0 & 1 & 0 & \ldots & 0 \\ 0 & 0 & 1 & \ldots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -\alpha_0 & -\alpha_1 & -\alpha_2 & \ldots & -\alpha_{n-1} \end{bmatrix}}_{\hat{A}} Y_i. \tag{5}$$

This equality comes from the construction $Y_i = \begin{bmatrix} y_i & \dot{y}_i & \ldots & y_i^{(n-1)} \end{bmatrix}^T$ where $\alpha_1, \alpha_2, \ldots, \alpha_{n-1}$ are the coefficients of the characteristic polynomial of A, i.e., $\det(\lambda I - A) = \lambda^n + \alpha_{n-1}\lambda^{n-1} + \cdots + \alpha_0$. This can be seen by nothing that $y_i^{(n)} = C_i A^n x = C_i(\alpha_{n-1}A^{n-1} + \cdots + \alpha_1 A + \alpha_0)x = \alpha_{n-1}y_i^{n-1} + \cdots + \alpha_1 \dot{y}_i + \alpha_0 y_i$ whenever node i is attack free. Moreover, we note that $\hat{A}$ is the controller form of A and has the same eigenvalues as A and by Assumption 6 all the eigenvalues of $\hat{A}$ have positive real parts.

Remark 1. *We note that the state portion corresponding to the stable eigenvalues will decay to zero with the elapse of time. This argues for why 6 can be dropped without loss of generality.* □

Based on Equation (5), we define the following local sanity check:

Local sanity check[2]: given measurement $y_i(t)$ of node i, we say node i passes the local sanity check if $\|\dot{Y}_i(t) - \hat{A}Y_i(t)\| \leq \epsilon$ for any $t > 0$, where ϵ is the error tolerance.

In practice, we ask all nodes in the network to constantly run the local sanity check. It is trivially seen that all attack-free nodes would pass the check at any time. On the other hand, if a node fails the local sanity check, then we immediately reach the conclusion that this node is under attack. Therefore, we assume that all the attack vectors e_i corresponding to attacked nodes are constructed so that the resulting measurements $Y_i(t)$ pass the local sanity check, i.e., $\|\dot{Y}_i(t) - \hat{A}Y_i(t)\| \leq \epsilon$.

The purpose of the local sanity check is to force the dynamics of Y_i to be governed by (5), including those from attacked nodes. We may exploit this additional knowledge of $d_{1i}Y_i$ (or ϕ_i) to achieve better tracking results.

VI. Solving the DRST Problem

We argued in Section V that the tracking of the compressed measurements $(D \otimes I_n)Y$ is intrinsically identical to a dynamic average consensus problem. We extend the results in [14] from the scalar-input case to the vector case and present such extension in the time-domain.

We obtain that each node updates its estimate of the average according to:

[2]Again, we note that $\dot{Y}_i$ cannot be computed in real systems and thus the local sanity check can only be implemented for the corresponding discrete-time version of the results.

$$\begin{cases} \dot{J}_i = & -\hat{A}J_i + 2\hat{A}\phi_i - 2k_I \sum_{j\in\mathcal{N}_i}(\eta_j - \eta_i), \\ \dot{b}_i = & \hat{A}b_i + k_I \sum_{j\in\mathcal{N}_i}(J_j - J_i), \\ \eta_i = & k_P b_i + k_I \sum_{j\in\mathcal{N}_i}(J_j - J_i), \end{cases} \tag{6}$$

where J_i is the estimate at node i of the average of the input signal[3] $\frac{1}{p}\sum_i \phi_i$, k_I, k_P are design parameters, b_i is an internal state of node i and η_i is an additional state of node i. The following theorem states that under suitable choices of k_I and k_P the estimate of each node v_i approaches the true state $\frac{1}{p}\sum_i \phi_i$.

Theorem 1. *Consider the average tracking algorithm in* (6) *where the input signal satisfies* $\dot{\phi}_i = \hat{A}\phi_i$ *for all* $i \in \{1, 2, \ldots, p\}$. *There exist constants* $k_{I0} > 0$ *and* $k_{P0} > 0$ *such that for any* $k_I \geq k_{I0}$ *and* $k_P \geq k_{P0}$ *the state estimate* J_i *at each node tracks the average of the input signals exponentially fast, i.e.:*

$$\left\| J_i(t) - \frac{1}{p^{\frac{3}{2}}} \sum_i \phi_i(t) \right\| < \beta e^{-\alpha t},$$

for some $\alpha, \beta > 0$. □

Before proving the theorem we present the following two lemmas that will be used afterwards.

Lemma 3. *Consider a positive semi-definite matrix* $B = B^T \in \mathbb{R}^{n\times n}$ *and a matrix* $S \in \mathbb{R}^{n\times m}$ *such that* $\mathcal{R}(S) \cap \ker(B) = \{0\}$. *The matrix* $S^T B S$ *is diagonalizable and positive definite.* □

Lemma 4. *Consider a block matrix* B *of the form* $B = \begin{bmatrix} -A - 2\lambda^2 I_n & -2c\lambda I_n \\ \lambda I_n & A \end{bmatrix}$ *where* $A \in \mathbb{R}^{n\times n}$. *There exist* $\lambda_0, c_0 > 0$ *such that* B *is Hurwitz for any* $\lambda > \lambda_0$ *and* $c > c_0$. □

Instead of proving directly Theorem 1, for the sake of simplicity, we prove the convergence result for a broader class of systems in the form of:

$$\begin{cases} \dot{v} = & -Ev + 2E\varphi - 2k_I F\eta, \\ \dot{m} = & Em + k_I F v, \\ \eta = & k_P m + k_I F v, \end{cases} \tag{7}$$

where $k_I, k_P \in \mathbb{R}$, $v, m, \varphi, \eta \in \mathbb{R}^a$ for some $a \in \mathbb{N}$, $E, F \in \mathbb{R}^{a\times a}$, $F = F^T$ is positive semi-definite and all eigenvalues of E have positive real parts, which implies that $\varphi(t)$ is an exponentially growing input satisfying $\dot{\varphi} = E\varphi$.

We first show that tracking algorithms of the form of (7) can be decoupled into 2 sub-systems and then we analyze the convergence properties for each sub-system. To do this, we introduce an orthogonal matrix $\begin{bmatrix} R & S \end{bmatrix} \in \mathbb{R}^{a\times a}$ and make the following two assumptions:

[3]In previous sections we denote this value by J_i^1 and the superscript is dropped here for simpler notation.

1) **Additional Assumption 1:** The matrices R and S have the properties that $\mathcal{R}(R) = \ker F$, $\mathcal{R}(S) \cap \ker F = \{0\}$, and $R^T E S = 0$, $S^T E R = 0$.
2) **Additional Assumption 2:** E and F are simultaneously block diagonalizable and each block in E corresponds to an identity block in F up to some scaling factor λ_i. In other words, there exists an invertible matrix P such that $P^{-1}EP = \text{diag}\{\Lambda_1, \ldots, \Lambda_r\}$ and $P^{-1}FP = \text{diag}\{\lambda_1 I_{s_1}, \ldots, \lambda_r I_{s_r}\}$ such that $\Lambda_i \in \mathbb{R}^{s_i \times s_i}$ with i ranging from 1 to r.

Note that these two additional assumptions are introduced to prove the desired tracking property and any system in the form of (6) automatically falls into the broader class of systems described in (7), which we will argue at the end of this section. Now we are ready to present our decomposition result based on the change of coordinates $(z_1, z_2) = (R^T, S^T)v$ where z_1 describes the sum of local estimates and z_2 is the difference among local estimates. We see that if z_1 converges to $R^T\varphi$ and z_2 converges to 0, then all nodes reach the consensus φ which is the target of the tracking algorithm.

Proposition 1. *The dynamical system in* (7) *can be decoupled into the following two sub-systems if it satisfies Additional Assumption 1:*

$$\dot{z}_1 = -R^T E R z_1 + 2R^T E\varphi, \tag{8}$$

and:

$$\begin{cases} \dot{z}_2 = & -S^T E S z_2 - 2k_I S^T F S f_2 + 2S^T E\varphi, \\ \dot{g}_2 = & S^T E S g_2 + k_I S^T F S z_2, \\ f_2 = & k_P g_2 + k_I S^T F S z_2, \end{cases} \tag{9}$$

where $z_1 = R^T v$, $z_2 = S^T v$, $f_2 = S^T\eta$ *and* $g_2 = S^T m$. □

The following proposition shows that z_1 converges to $R^T\varphi$ as desired.

Proposition 2. *For any solution* $z_1(t)$ *of* (8) *we have:*

$$\left\| z_1(t) - R^T\varphi(t) \right\| < \beta e^{-\alpha t}$$

for some $\alpha, \beta > 0$. □

Similarly, we prove that z_2 converges to 0 in the following proposition.

Proposition 3. *For any solution* $(z_2(t), g_2(t))$ *of* (9) *we have:*

$$\|z_2(t)\| < \beta e^{-\alpha t},$$

for some $\alpha, \beta > 0$ *if the system defined in* (7) *satisfies additional assumptions 1 and 2.* □

Lastly, we go back to prove Theorem 1. The dynamic average tracking algorithm in (6) can be compactly written in the form:

$$\begin{cases} \dot{J} = & -(I_p \otimes A)J + 2(I_p \otimes A)\phi - 2k_I(\mathcal{L} \otimes I_n)\eta, \\ \dot{b} = & (I_p \otimes A)b + k_I(\mathcal{L} \otimes I_n)J, \\ \eta = & k_P b + k_I(\mathcal{L} \otimes I_n)J, \end{cases} \tag{10}$$

where $J = \begin{bmatrix} J_1^T & J_2^T & \dots & J_p^T \end{bmatrix}^T$ is obtained by concatenating estimates from all nodes, $b = \begin{bmatrix} b_1^T & b_2^T & \dots & b_p^T \end{bmatrix}^T$ and $\eta = \begin{bmatrix} \eta_1^T & \eta_2^T & \dots & \eta_p^T \end{bmatrix}^T$ are similarly defined. Recall that $\mathcal{L}$ is the Laplacian of the graph and $\otimes$ denotes the kronecker product. It is not hard to check that system (10) falls into the class of systems of the form (7) by picking $E = I_p \otimes A$ all of whose eigenvalues have positive real parts, $F = F^T = \mathcal{L} \otimes I_n \succeq 0$, and $a = np$. We also pick $\varphi = \begin{bmatrix} \phi_1^T & \phi_2^T & \dots & \phi_p^T \end{bmatrix}^T$ and $\dot{\phi}_i = A\phi_i$ with i ranging from 1 to p and obtain that $\dot{\varphi} = (I_p \otimes A)\varphi = E\phi$ which meets the requirement.

Next we show that the additional assumptions 1 and 2 are satisfied. The conditions $\mathcal{R}(R) = \ker F$ and $\mathcal{R}(S) \cap \ker F = \{0\}$ are naturally satisfied by the construction of R and S. We also notice that $\mathcal{L}$ is the Laplacian of the graph and the only eigenvector corresponding to the zero eigenvalue is $\mathbf{1}_p$. Therefore, we pick $R = \frac{1}{\sqrt{p}}\mathbf{1}_p$ and $\begin{bmatrix} R & S \end{bmatrix}$ being an orthogonal matrix. We note that $\begin{bmatrix} R & S \end{bmatrix} \otimes I_n$ is also an orthogonal matrix with $R \otimes I_n$ lying in the kernel of $(\mathcal{L} \otimes I_n)$. Moreover, we can check by direct computation that the following equalities hold:

$$\begin{aligned}(R^T \otimes I_n)E(S \otimes I_n) &= (R^T \otimes I_n)(I_p \otimes A)(S \otimes I_n) \\ &= R^T S \otimes A = 0, \\ (S^T \otimes I_n)E(R \otimes I_n) &= (S^T \otimes I_n)(I_p \otimes A)(R \otimes I_n) \\ &= S^T R \otimes A = 0,\end{aligned}$$

which satisfy the additional assumption 1. Also, $\mathcal{L} = \mathcal{L}^T$ for an undirected graph, $\mathcal{L}$ is diagonalizable and we assume $T^{-1}\mathcal{L}T$ is a diagonal matrix with eigenvalues $\lambda_1, \dots, \lambda_r$. To show that the additional assumption 2 is also satisfied, we directly compte the following two similarity transformations: $(T \otimes I_n)^{-1}(\mathcal{L} \otimes I_n)(T \otimes I_n) = \text{diag}\{\lambda_1 I_n, \dots, \lambda_r I_n\}$ and $(T \otimes I_n)^{-1}(I_p \otimes A)(T \otimes I_n) = I_p \otimes A = \text{diag}\{A, \dots, A\}$. This calculation shows that F and E are simultaneously diagonalizable and each block in E corresponds to an identity block in F, after diagonalization, up to a scalar λ_i, hence the additional assumption 2 is also satisfied.

Since the dynamic average tracking system in (10) falls into the class (7) and satisfies both additional assumptions, we can apply Propositions 2 and (3) to analyze the trajectory of J_is in Equation (6). By Proposition 2 we conclude that $(R^T \otimes I_n)J = \frac{1}{\sqrt{p}}\sum_i J_i$ converges to $(R^T \otimes I_n)\varphi = \frac{1}{\sqrt{p}}\sum_i \phi_i$ exponentially fast. By Proposition 3 we conclude $(S^T \otimes I_n)J$ converges to 0 exponentially fast, which shows the difference between any two estimates in two nodes decays to zero exponentially fast. Combining both we note any J_i converges to $\frac{1}{p}\sum_i \phi_i$ and hence Theorem 1 is proved.

At this point, we know that if each node in the network executes the tracking algorithm (6), then its estimate vector J_i converges to $(D \otimes I_n)Y$ exponentially. The missing piece now is a decoding algorithm, that enables each node to reconstruct the state estimate $\hat{x}_i(t)$ from the compressed measurement J_i. However, in the interest of space we will not present the decoding algorithm, but proceed to the main result of this paper:

Theorem 2. *Consider the linear system subject to attacks defined in* (1) *satisfying assumptions 1-6, the tracking algorithm* (6) *together with a state-reconstruction algorithm enables each node to asymptotically track the state of the system* (1)*, if the pair* (A, C) *is 2s-sparse detectable.* □

VII. Conclusion

In this paper, we propose a novel algorithm that solves the decentralized resilient state-tracking problem. The proposed solution consists of a dynamic consensus algorithm enabling each node to compute a compressed version of all the measurements that it then use by any secure state reconstruction algorithm to track the state. Compared with existing solutions of the DRST problem, our algorithm solves the DRST problem for a wider class of systems than any of the solutions available in the literature.

References

[1] J. P. Farwell and R. Rohozinski, "Stuxnet and the future of cyber war," *Survival*, vol. 53, no. 1, pp. 23–40, 2011.

[2] "State farm falls victim to credential-stuffing attack," https://threatpost.com/state-farm-credential-stuffing-attack/147139/, accessed: 2020-12-23.

[3] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure state-estimation for dynamical systems under active adversaries," in *2011 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2011, pp. 337–344.

[4] Y. Mao, A. Mitra, S. Sundaram, and P. Tabuada, "When is the secure state-reconstruction problem hard?" in *2019 IEEE 58th Conference on Decision and Control (CDC)*, 2019, pp. 5368–5373.

[5] Y. Shoukry and P. Tabuada, "Event-triggered state observers for sparse sensor noise/attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2079–2091, 2015.

[6] Y. Shoukry, M. Chong, M. Wakaiki, P. Nuzzo, A. Sangiovanni-Vincentelli, S. A. Seshia, J. P. Hespanha, and P. Tabuada, "Smt-based observer design for cyber-physical systems under sensor attacks," *ACM Transactions on Cyber-Physical Systems*, vol. 2, no. 1, pp. 1–27, 2018.

[7] M. S. Chong, M. Wakaiki, and J. P. Hespanha, "Observability of linear systems under adversarial attacks," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2439–2444.

[8] A. Mitra and S. Sundaram, "Byzantine-resilient distributed observers for LTI systems," *Automatica*, vol. 108, p. 108487, 2019.

[9] Y. Mao, A. Mitra, S. Sundaram, and P. Tabuada, "On the computational complexity of the secure state-reconstruction problem," *arXiv preprint arXiv:2101.01827*, 2021.

[10] J. G. Lee, J. Kim, and H. Shim, "Fully distributed resilient state estimation based on distributed median solver," *IEEE Transactions on Automatic Control*, vol. 65, no. 9, pp. 3935–3942, 2020.

[11] J. G. Lee and H. Shim, "A tool for analysis and synthesis of heterogeneous multi-agent systems under rank-deficient coupling," *Automatica*, vol. 117, p. 108952, 2020.

[12] X. He, X. Ren, H. Sandberg, and K. Johansson, "How to secure distributed filters under sensor attacks?" *ArXiv*, vol. abs/2004.05409, 2020.

[13] S. S. Kia, B. Van Scoy, J. Cortes, R. A. Freeman, K. M. Lynch, and S. Martinez, "Tutorial on dynamic average consensus: The problem, its applications, and the algorithms," *IEEE Control Systems Magazine*, vol. 39, no. 3, pp. 40–72, 2019.

[14] H. Bai, R. A. Freeman, and K. M. Lynch, "Robust dynamic average consensus of time-varying inputs," in *49th IEEE Conference on Decision and Control (CDC)*, 2010, pp. 3104–3109.

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

[16] P. J. Antsaklis and A. N. Michel, *Linear systems*. Springer Science & Business Media, 2007.

[17] Y. Mao and P. Tabuada, "Decentralized resilient state-tracking," Technical report, 2021. [Online]. Available: https://sites.google.com/a/cyphylab.ee.ucla.edu/cyber-physical-systems-laboratory/Home/publications/UCLA-CyPhyLab-2021-09.pdf