53rd IEEE Conference on Decision and Control
December 15-17, 2014. Los Angeles, California, USA

Secure state estimation and control using multiple (insecure) observers

Shaunak Mishra, Nikhil Karamchandani, Paulo Tabuada and Suhas Diggavi

Abstract— **Motivated by the need to protect Cyber-Physical Systems against attacks, we consider in this paper the problem of estimating the state in a private and secure manner despite active adversary attacks; adversaries that can attack the software/hardware where state estimation is performed. To combat such threats, we propose an architecture where state estimation is performed across multiple computing nodes (observers). We then show that even when ρ out of a total $3\rho+1$ observers are actively attacked: 1) using a combination of outputs from the observers, the state is still correctly estimated; 2) the physical plant is still correctly controlled; 3) the adversary can only obtain limited knowledge about the state. Our approach is inspired by techniques in cryptography for secure message transmission and information-theoretic secrecy. In addition, our guarantees on the secrecy of the plant's state against corrupting observers are based on the Cramer-Rao lower bound from estimation theory.**

I. Introduction

The security of Cyber-Physical Systems (CPSs) is a problem of increasing importance as we discover that much of the critical infrastructure we depend on is vulnerable to cyber attacks [1], [2], [3]. While it can be argued that many CPSs are physically secure in the sense that maliciously intended people cannot gain physical proximity, they can still be remotely attacked [4]. CPSs that are remotely operated, such as Unmanned Aerial Vehicles (UAVs) and parts of the power grid, can be vulnerable to several attack methodologies, since most of these systems rely on complex control algorithms running on networked digital platforms. For example, this could be enabled by hardware malware in the chips used in these platforms that becomes active at the discretion of an attacker [5].

In this paper, we are concerned with attackers that want to control a CPS in order to alter its normal operation. We have two objectives: (i) control the plant correctly, and (ii) prevent the adversary from learning the plant's state. When state estimates are computed in a single location, an adversary which has access to that location (through hardware or software malware) could use the state estimate for initiating attacks. Therefore, we propose an architecture where state estimation is distributed across several computing nodes (observers), as shown in Figure 1 (discussed later in Section II). The challenge is to perform accurate state estimation for controlling the CPS, despite an attacker which has access to a fraction of these observers. In this paper, we present a solution to this problem and prove that even when ρ out of $3\rho+1$ observers are arbitrarily corrupted, we can still operate the CPS correctly, *i.e.,* we can still control the system as desired and prevent the adversary from learning the state to any desired accuracy.

Our solution is inspired by *secure message transmission* (SMT) [6], a problem studied in cryptography for finite fields. In this problem, a message is securely transmitted between two agents, despite an active adversary who partially controls the communication channels. The main differences in our setup are two-fold: (i) we operate over reals rather than finite fields. This means that it is not possible to give perfect secrecy guarantees and therefore we formulate secrecy as an estimation error guarantee for any adversary. We also give guarantees against a strong active adversary who has complete knowledge of the system parameters and has unbounded power (both transmit and computational). (ii) The SMT problem is posed in a static context, where a given message is to be transmitted. On the other hand, the control and state estimates dynamically change over time in our setup due to the dynamics of a physical plant, and we need to perform these dynamic computations securely. Our techniques are informed by making a connection between our problem and algebraic real error correction (through Reed-Solomon codes [7]) and estimation theory. For simplicity, in this paper we focus on the case where there is no measurement and actuator noise. However, the ideas can be easily extended for this case, since we ensure that the original state estimate based on the plant output is reconstructed in a distributed and secure manner.

The problem of adversarial attacks in multi-agent networks has been studied in several contexts, for example distributed consensus [8], [9] and function computation [10], [11]. Our goal is not consensus or distributed function computation, but reliable control of a physical plant despite adversarial attacks. Although consensus problems and distributed function computation through linear iterative strategies also involve dynamics, we consider arbitrary linear plants and thus cannot design the dynamics as is possible in these problems. Differential private filtering, studied in [12], consists of a methodology to protect the identity of the nodes contributing information. In our case we seek to protect, not the identity, but the state. In [13] the problem of securing the state of the plant from a passive adversary is studied; in contrast, we allow an active adversary who can also disrupt the legitimate state estimation and control, and our security requirement also differs from their setup.

The remainder of this paper is organized as follows. Section II describes the problem setup, system architecture, and notation. Next, we illustrate our key ideas for the case where the adversary attacks a single observer, with Sections III and IV focusing on a passive and active adversary

The authors are with the Electrical Engineering Department, University of California, Los Angeles. The work was supported by NSF award 1136174.

978-1-4673-6090-6/14/$31.00 ©2014 IEEE

respectively. In Section V, we extend our results to an active adversary controlling ρ observers and demonstrate how we can operate correctly despite adversarial corruptions when we use at least $3\rho+1$ observers.

II. Notation and Setup

We first describe the model for plant dynamics and then introduce the proposed multiple observer setup. This is followed by the adversary model in the multiple observer setup and the constraints for the plant's operation in the presence of such an adversary.

A. Plant dynamics

The plant is modeled as a linear time invariant system as shown below:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \quad \mathbf{y}(t) = \mathbf{C}\mathbf{x}(t) \tag{1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ is the plant's state at time t, $\mathbf{u}(t) \in \mathbb{R}^m$ is input to the plant at time t, and $\mathbf{y}(t) \in \mathbb{R}^p$ is the plant's output at time t. For simplicity, in the usual setting (without security constraints), we consider a Luenberger observer [14] for estimating the state of the plant. The Luenberger observer receives the plant's input and output (*i.e.,* $\mathbf{u}(t)$ and $\mathbf{y}(t)$) and uses the following update rule for the state estimate:

$$\hat{\mathbf{x}}(t+1) = \mathbf{A}\hat{\mathbf{x}}(t) + \mathbf{B}\mathbf{u}(t) + \mathbf{L}\left(\mathbf{y}(t) - \mathbf{C}\hat{\mathbf{x}}(t)\right) \tag{2}$$

where $\hat{\mathbf{x}}(t) \in \mathbb{R}^n$ is the observer's state estimate at time t and L is the observer gain. The state estimate $\hat{\mathbf{x}}(t)$ from the observer is used with the *external* reference command $\mathbf{r}(t) \in \mathbb{R}^m$ (discussed in Section II-E) and a local stabilizing controller with gain matrix K, resulting in the control law:

$$\mathbf{u}(t) = \mathbf{r}(t) + \mathbf{K}\hat{\mathbf{x}}(t). \tag{3}$$

In the remainder of this paper, we will refer to the setup defined by (1), (2), and (3) as the single observer setup.

Throughout the paper we make the simplifying assumption that the observer estimate $\hat{\mathbf{x}}$ at time $t=0$ equals the state $\mathbf{x}$ at time $t=0$. Although counterintuitive, this results in no loss of generality since the secrecy and security guarantees we provide under this assumption extend to the case where $\hat{\mathbf{x}}(0) \neq \mathbf{x}(0)$ (see [15] for details). Under this assumption, the plant dynamics can be simplified as follows:

$$\mathbf{x}(t+1) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}\left(\mathbf{K}\mathbf{x}(t) + \mathbf{r}(t)\right) = \mathbf{A}_{cl}\mathbf{x}(t) + \mathbf{B}\mathbf{r}(t) \tag{4}$$

where $\mathbf{A}_{cl} = \mathbf{A} + \mathbf{B}\mathbf{K}$. Without loss of generality, we assume that $\mathbf{x}(0) = \mathbf{0}$ (initial state of the plant). Hence, given a sequence of inputs $\mathbf{r}(0), \mathbf{r}(1), \ldots \mathbf{r}(l-1)$, the sequence of plant states can be written as follows:

$$\begin{bmatrix} \mathbf{x}(1) \\ \mathbf{x}(2) \\ \vdots \\ \mathbf{x}(l) \end{bmatrix} = \begin{bmatrix} \mathbf{B} & \mathbf{0} & \ldots & \mathbf{0} \\ \mathbf{A}_{cl}\mathbf{B} & \mathbf{B} & \ldots & \mathbf{0} \\ \vdots & \vdots & \ddots & \mathbf{0} \\ \mathbf{A}_{cl}^{l-1}\mathbf{B} & \mathbf{A}_{cl}^{l-2}\mathbf{B} & \ldots & \mathbf{B} \end{bmatrix} \begin{bmatrix} \mathbf{r}(0) \\ \mathbf{r}(1) \\ \vdots \\ \mathbf{r}(l-1) \end{bmatrix} = \mathbf{J}_l \mathbf{r}_{0:l-1}. \tag{5}$$

As shown above, we use the notation $\mathbf{r}_{t_1:t_2}$ for $\left[\mathbf{r}^T(t_1) \quad \mathbf{r}^T(t_1+1) \quad \ldots \quad \mathbf{r}^T(t_2)\right]^T$ where T denotes matrix transposition.

B. Multiple observer setup

In the multiple observer setup, the state observer, as shown in (2), is *distributed* among multiple computing nodes. Figure 1 shows the multiple observer setup (with d observers). The external reference input $\mathbf{r}(t)$ and plant output $\mathbf{y}(t)$ are sent to encoders ENC_I and ENC_O, respectively, as opposed to being directly sent to the observers. Observer $i \in \{1,2,\ldots,d\}$ receives at time t an encoded version of $\mathbf{r}(t)$, denoted by $\mathbf{r}_i(t)$, from ENC_I and an encoded version of $\mathbf{y}(t)$, denoted by $\mathbf{y}_i(t)$ from ENC_O. In the absence of any adversarial corruptions, the state estimate update rule for observer i is as shown below:

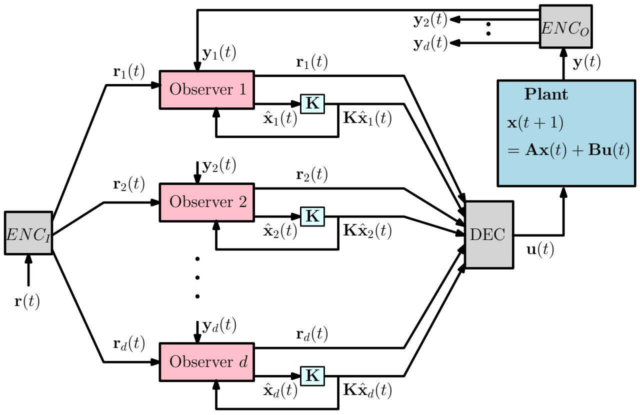

Fig. 1. A d-observer setup for state estimation.

$$\hat{\mathbf{x}}_i(t+1) = \mathbf{A}\hat{\mathbf{x}}_i(t) + \mathbf{B}\left(\mathbf{K}\hat{\mathbf{x}}_i(t) + \mathbf{r}_i(t)\right) + \mathbf{L}\left(\mathbf{y}_i(t) - \mathbf{C}\hat{\mathbf{x}}_i(t)\right) \tag{6}$$

where $\hat{\mathbf{x}}_i(t)$ is the state estimate of observer i at time t. Clearly, the above update rule is similar to (2) in the single observer setup; the main difference lies in using $\mathbf{r}_i(t)$ and $\mathbf{y}_i(t)$ instead of $\mathbf{r}(t)$ and $\mathbf{y}(t)$.

In the absence of any adversarial corruptions, the decoder *DEC* receives $\mathbf{r}_i(t)$ and $\mathbf{K}\hat{\mathbf{x}}_i(t)$ for $i \in \{1,2,\ldots,d\}$ as shown in Figure 1. The number d of observers and the design of ENC_I, ENC_O, and *DEC* is based on the specifications (described in Section II-C) of the adversary who can corrupt a fraction of the observers. We assume that the encoders have access to random number generators and there is no shared randomness between the encoders and the decoder.

C. Adversary model

We now describe the adversary model in the context of the multiple observer setup described in Section II-B. In this paper, we consider two types of adversaries: passive and active. The difference between these two types is in the nature of adversarial behavior.

Passive adversary: A ρ-passive adversary can tap into any subset of ρ observers in a d-observer setup and access all the inputs to the particular subset of observers. Such adversaries are also referred to as *honest-but-curious* in the cryptography literature since they do not affect the normal operation of a protocol but just try to infer useful information. In the multiple observer setup, the objective of a ρ-passive adversary is to estimate useful information such as the plant's state sequence or the reference input sequence based on inputs to the set of tapped observers.

Active adversary: A ρ-active adversary is more powerful than a ρ-passive adversary. It not only has access to all the inputs to the set of affected observers (any ρ observers in a d-observer setup), but can also *inject* errors (of arbitrary magnitude) in the outputs of attacked observers. Furthermore, the adversary can also alter the internal operations (*e.g.,* state estimate update rule) of the attacked observers. Since the outputs from the observers influence the input to the plant, an active adversary can potentially alter the normal operation of the plant.

In both the cases (passive and active), the adversary has unbounded computational power. It also has knowledge of the plant parameters, and the operations done by ENC_I, ENC_O, and DEC. The adversary does not have access to the random number generators in the input encoder (ENC_I) and output encoder (ENC_O); this is essentially the source of secrecy in the multiple observer setup.

D. Constraints: correctness and secrecy

In a d-observer setup, with initial plant state $\mathbf{x}(0) = \mathbf{0}$ (known to the adversary) and external reference input sequence $\mathbf{r}_{0:l-1}$ (unknown to the adversary) we consider the following constraints:

a) Correctness: The evolution of the plant in the d-observer setup is exactly the same as in the single observer setup; even in the presence of an active adversary which can arbitrarily change the outputs from the set of attacked observers. Formally, for any given input sequence $\mathbf{r}_{0:l-1}$, the plant's state sequence is $\mathbf{x}_{1:l} = \mathbf{J}_l \mathbf{r}_{0:l-1}$ (as shown in (5) for the single observer setup with no adversary) despite the attack of an active adversary.

b) Secrecy: An adversary (ρ-active or ρ-passive) having access to the inputs of any ρ observers should have limited knowledge of the external reference input sequence $\mathbf{r}_{0:l-1}$ and plant's state sequence $\mathbf{x}_{1:l}$. Formally, if $\mathbf{E}_{r,0:l-1}$ and $\mathbf{E}_{x,1:l}$ are the error covariance matrices corresponding to the adversary's estimate of $\mathbf{r}_{0:l-1}$ and $\mathbf{x}_{1:l}$, then the following should be satisfied:

$$\mathrm{tr}(\mathbf{E}_{r,0:l-1}) > \phi_r > 0, \quad \mathrm{tr}(\mathbf{E}_{x,1:l}) > \phi_x > 0 \tag{7}$$

where $\mathrm{tr}(\cdot)$ denotes the matrix trace operation, and ϕ_r and ϕ_x are constant design parameters which can be adjusted for any desired level of secrecy. It should be noted that since we assume $\hat{\mathbf{x}}(0) = \mathbf{x}(0)$, $\mathbf{x}(0)$ is known to each observer; but the encoded inputs and encoded outputs are responsible for the observer's uncertainty about $\mathbf{r}_{0:l-1}$ and $\mathbf{x}_{1:l}$.

An important aspect of the d-observer setup is the minimum number d_{min} of observers required to ensure the constraints mentioned above against an adversary (ρ-active or ρ-passive). Clearly, d_{min} depends on ρ, and whether the adversary is active or passive. Using arguments similar to [6], it can be easily shown that $d_{min} \geq \rho + 1$ for a ρ-passive adversary, and $d_{min} \geq 3\rho + 1$ for a ρ-active adversary.

E. Discussion

The described setup is appropriate for Cyber-Physical Systems that are remotely operated. A case in point are Unmanned Air Vehicles (UAV) where state estimation and the computation of local controllers is performed onboard while the reference input $\mathbf{r}$ is remotely sent by a pilot. Another typical example are SCADA systems where local observers and controllers regulate different physical quantities based on set points that are remotely sent from a central supervisor. In all of these scenarios, we envision attacks on the communication between the operator and the local observers/controllers and between the local observers/controllers and the actuators. We also envision either software or hardware attacks on the observers/local controllers. We exclude from our model attacks on the actuators since an attacker that can command an actuator can immediately cause damage to the plant. Hence, actuators need to be physically hardened to withstand attacks. We also exclude attacks to the operator since in many situations, e.g., UAVs, it is located in a secure facility.

III. 1-PASSIVE ADVERSARY

As mentioned in Section II-D, $d_{min} \geq 2$ for a 1-passive adversary. In this section, we show that $d_{min} = 2$ for a 1-passive adversary by designing a 2-observer setup (in Section III-A) and showing that the correctness and secrecy constraints are satisfied (in Sections III-B and III-C respectively).

A. 2-observer setup

The operations of the encoders, observers (indexed by i), and decoder in the 2-observer setup are described below.

Encoder: The following operations are done at the input encoder ENC_I which receives $\mathbf{r}(t)$ as input:

$$\mathbf{r}_1(t) = \frac{\mathbf{r}(t)}{2} + \boldsymbol{\theta}(t), \quad \mathbf{r}_2(t) = \frac{\mathbf{r}(t)}{2} - \boldsymbol{\theta}(t) \tag{8}$$

where $\boldsymbol{\theta}(t) \in \mathbf{R}^m$ is a random vector drawn from a multivariate Gaussian distribution with zero mean and covariance matrix $\sigma^2\mathbf{I}_m$ ($\mathbf{I}_m$ is the identity matrix of dimension m and σ is a positive real number). In the remainder of this paper, we use the notation $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to denote the multivariate Gaussian distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. It should be noted that $\boldsymbol{\theta}(t)$ is intentionally generated by the input encoder ENC_I and is i.i.d. (independent and identically distributed) over time. Similarly to ENC_I, the output encoder ENC_O receives $\mathbf{y}(t)$ as input and performs the following operations:

$$\mathbf{y}_1(t) = \frac{\mathbf{y}(t)}{2} + \boldsymbol{\delta}(t), \quad \mathbf{y}_2(t) = \frac{\mathbf{y}(t)}{2} - \boldsymbol{\delta}(t) \tag{9}$$

where $\boldsymbol{\delta}(t)$ is intentionally generated random vector $\sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}_p\right)$ and is i.i.d. over time. We justify the use of the Gaussian distribution for $\boldsymbol{\theta}(t)$ and $\boldsymbol{\delta}(t)$ while analyzing the secrecy constraint in Section III-C. Moreover, the random vectors generated by ENC_O and ENC_I are assumed to be independent.

Observer: For $i \in \{1,2\}$, observer i receives $\mathbf{r}_i(t)$ and $\mathbf{y}_i(t)$ at time t, and uses update rule (6) for its state estimate $\hat{\mathbf{x}}_i(t)$. Recall that we assume that observer i has knowledge of $\mathbf{x}(0)$ and thus sets its initial state estimate as $\hat{\mathbf{x}}_i(0) = \frac{\mathbf{x}(0)}{2}$.

Decoder: For $i \in \{1,2\}$, the decoder receives $\mathbf{K}\hat{\mathbf{x}}_i(t)$ and $\mathbf{r}_i(t)$ at time t, and simply adds all its inputs to obtain $\mathbf{u}(t)$ (fed to the plant) as shown below:

$$\mathbf{u}(t) = (\mathbf{K}\hat{\mathbf{x}}_1(t) + \mathbf{r}_1(t)) + (\mathbf{K}\hat{\mathbf{x}}_2(t) + \mathbf{r}_2(t)) \ . \tag{10}$$

B. Correctness

For correctness, given any external reference input sequence $\mathbf{r}_{0:l-1}$, we need the plant's state sequence $\mathbf{x}_{1:l}$ to be exactly as shown in (5). We prove the following claim, which is sufficient to show correctness.

Claim 1: Assuming the operations of ENC_I, ENC_O, DEC, and observers are as described in Section III-A, the following is true for all $t \geq 0$:

$$\hat{\mathbf{x}}_1(t+1) + \hat{\mathbf{x}}_2(t+1) = \mathbf{A}_{cl}\mathbf{x}(t) + \mathbf{B}\mathbf{r}(t) = \mathbf{x}(t+1). \tag{11}$$

Proof: The proof follows by induction (see [15] for details). ■

Since $\mathbf{x}(t+1) = \mathbf{A}_{cl}\mathbf{x}(t) + \mathbf{B}\mathbf{r}(t)$, the plant's state sequence $\mathbf{x}_{1:l}$ given input sequence $\mathbf{r}_{0:l-1}$ is exactly as shown in (5).

C. Secrecy

In order to perform the secrecy analysis we start by listing the observations of a 1-passive adversary (we consider the case when the 1-passive adversary taps observer 1; the analysis for observer 2 follows by symmetry). The adversary knows the initial state estimate $\hat{\mathbf{x}}_1(0) = \frac{\mathbf{x}(0)}{2} = \mathbf{0}$ and observes, up to time l, the sequence of encoded reference inputs $\mathbf{r}_1(0), \mathbf{r}_1(1), \dots \mathbf{r}_1(l-1)$ and the sequence of encoded sensor measurements $\mathbf{y}_1(1), \mathbf{y}_1(2), \dots \mathbf{y}_1(l)$ fed to observer 1. Hence, the information available to the adversary can be summarized as the vector $\mathbf{v}_l$:

$$\mathbf{v}_l \overset{(a)}{=} \begin{bmatrix} (\mathbf{I}_l \otimes \mathbf{C})\,\mathbf{J}_l \\ \mathbf{I}_{ml} \end{bmatrix} \mathbf{r}_{0:l-1} + 2\begin{bmatrix} \boldsymbol{\delta}_{1:l} \\ \boldsymbol{\theta}_{0:l-1} \end{bmatrix} = \mathbf{H}_l\mathbf{r}_{0:l-1} + \mathbf{z}_l \tag{12}$$

where (a) follows from correctness, proved in Section III-B ($\mathbf{J}_l$ is defined in (5)), and $\otimes$ denotes Kronecker product. Equation (12) shows that the adversary's observations are affine on the reference input $\mathbf{r}$. The the adversary's objective is then to estimate $\mathbf{r}_{0:l-1}$. Note that $\mathbf{z}_l$ is unknown to the adversary although it knows the distribution from which the elements of $\mathbf{z}_l$ are drawn. In this context, there can be several choices for the estimation criterion (*e.g.,* biased or unbiased) [16], [17]. For concreteness, in this paper we give guarantees on the accuracy of a minimum variance unbiased (MVU) estimate [16] made by the adversary; the guarantees can be easily extended for biased estimators using results in [17]. Given (12), the accuracy of the adversary's MVU estimate of $\mathbf{r}_{0:l-1}$ is fundamentally limited by the Cramer-Rao lower bound (CRLB) [16]. The CRLB for the affine model (12) can be easily evaluated (see [16] for details) as shown below:

$$\mathbf{E}_{r,0:l-1} \succeq \left(\mathbf{H}_l^T \boldsymbol{\Sigma}_z^{-1} \mathbf{H}_l\right)^{-1} = 4\sigma^2 \left(\mathbf{H}_l^T \mathbf{H}_l\right)^{-1} \tag{13}$$

where $\mathbf{E}_{r,0:l-1}$ is the error covariance matrix for the adversary's MVU estimate of $\mathbf{r}_{0:l-1}$, and $\boldsymbol{\Sigma}_z$ is the covariance matrix of $\mathbf{z}_l$ (in (12)). The above result also implies that the trace of $\mathbf{E}_{r,0:l-1}$ is not less than $4\sigma^2 \mathrm{tr}\left(\left(\mathbf{H}_l^T\mathbf{H}_l\right)^{-1}\right)$.

The plant's state sequence $\mathbf{x}_{1:l}$ is the linear function $\mathbf{x}_{1:l} = \mathbf{J}_l\mathbf{r}_{0:l-1}$ of the input sequence. Hence, the CRLB for $\mathbf{x}_{1:l}$ can be derived from the CRLB for $\mathbf{r}_{0:l-1}$ [16] as shown below:

$$\mathbf{E}_{x,1:l} \succeq 4\sigma^2 \mathbf{J}_l \left(\mathbf{H}_l^T\mathbf{H}_l\right)^{-1} \mathbf{J}_l^T. \tag{14}$$

Equations (13) and (14) show that by suitably adjusting σ we can impose any desired lower bound on the accuracy of the reference input and state estimates made by the adversary. Therefore, the secrecy constraint defined in (7) is satisfied. As a final remark we note that the Gaussian distribution is the best choice to generate the vector $\mathbf{z}_l$ since it is shown in [18] that it leads to the *worst* CRLB for an MVU estimator.

IV. 1-ACTIVE ADVERSARY

As mentioned in Section II-D, $d_{min} \geq 4$ is necessary for a 1-active adversary. In this section, we show that $d_{min} = 4$ is sufficient for a 1-active adversary by designing a 4-observer setup (in Section IV-A) and showing that the correctness and secrecy constraints are satisfied (in Sections IV-B and IV-C respectively).

A. 4-*observer setup*

The operations of the encoders, observers (indexed by i) and decoder in the 4-observer setup are described below.

Encoders: For $i \in \{1,2,3,4\}$, the following operation is done at the input encoder ENC_I which receives $\mathbf{r}(t)$ as input:

$$\mathbf{r}_i(t) = \mathbf{r}(t) + \lambda_i \boldsymbol{\theta}(t) \tag{15}$$

where $\boldsymbol{\theta}(t) \in \mathbb{R}^m$ is a random vector $\sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}_m\right)$ generated by ENC_I and is distributed i.i.d. over time. The scaling factor $\lambda_i \in \mathbb{R} - \{0\}$ is the same for all time t for observer i. Similarly, the following operation is done at the output encoder ENC_O:

$$\mathbf{y}_i(t) = \mathbf{y}(t) + \lambda_i \boldsymbol{\delta}(t) \tag{16}$$

where $\boldsymbol{\delta}(t) \in \mathbb{R}^p$ is a random vector $\sim \mathcal{N}\left(\mathbf{0}, \sigma^2\mathbf{I}_p\right)$ generated by ENC_O and is distributed i.i.d. over time. The scaling factor λ_i is same as the one used by ENC_I for observer i. The adversary is assumed to have knowledge of the scaling factor λ_i for each observer. Also, $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are assumed to be distinct (needed for proving correctness in Section IV-B). The random vectors generated by ENC_I and ENC_O are assumed to be independent.

Observers: The operations done at an observer which is not under the influence of an adversary are described below. For $i \in \{1,2,3,4\}$, observer i receives $\mathbf{r}_i(t)$ and $\mathbf{y}_i(t)$ at time t and uses update rule (6) for its state estimate $\hat{\mathbf{x}}_i(t)$. Since we assume that $\hat{\mathbf{x}}(0) = \mathbf{x}(0)$ observer i sets its initial state estimate as $\hat{\mathbf{x}}_i(0) = \mathbf{x}(0)$. However, a 1-active adversary can attack any of the observers and arbitrarily change its operation.

Decoder: For $i \in \{1,2,3,4\}$, the decoder *DEC* receives $\tilde{\mathbf{r}}_i(t)$ and $\tilde{\mathbf{k}}_i(t)$ at time t. Under normal operation (with no adversarial errors) $\tilde{\mathbf{r}}_i(t) = \mathbf{r}_i(t)$ and $\tilde{\mathbf{k}}_i(t) = \mathbf{K}\hat{\mathbf{x}}_i(t)$. When an adversary injects errors in the outputs of observer i, the decoder receives $\tilde{\mathbf{r}}_i(t) = \mathbf{r}_i(t) + \mathbf{e}_{i,r}(t)$ and $\tilde{\mathbf{k}}_i(t) = \mathbf{K}\hat{\mathbf{x}}_i(t) + \mathbf{e}_{i,k}(t)$, where $\mathbf{e}_{i,r}(t)$ and $\mathbf{e}_{i,k}(t)$ are errors (of arbitrary magnitude) introduced by the adversary. In this 1-active adversary setting, the decoder does not know a priori which observer is under the adversary's influence. Having received $\tilde{\mathbf{r}}_i(t)$ and $\tilde{\mathbf{k}}_i(t)$, the decoder computes the following for all pairs (i,i') such that $i,i' \in \{1,2,3,4\}$ and $i < i'$:

$$\mathbf{s}_{ii',r}(t) = \frac{\lambda_{i'}}{\lambda_{i'} - \lambda_i}\tilde{\mathbf{r}}_i(t) - \frac{\lambda_i}{\lambda_{i'} - \lambda_i}\tilde{\mathbf{r}}_{i'}(t) \tag{17}$$

$$\mathbf{s}_{ii',k}(t) = \frac{\lambda_{i'}}{\lambda_{i'} - \lambda_i}\tilde{\mathbf{k}}_i(t) - \frac{\lambda_i}{\lambda_{i'} - \lambda_i}\tilde{\mathbf{k}}_{i'}(t). \tag{18}$$

There are $\binom{4}{2} = 6$ possible $\mathbf{s}_{ii',r}(t)$ and the majority value (most frequently occurring) among these is denoted by $\mathbf{s}_r^*(t)$. Similarly, the majority value for $\mathbf{s}_{ii',k}(t)$ is denoted by $\mathbf{s}_k^*(t)$. We show in Section IV-B that the majority value for both $\mathbf{s}_{ii',r}(t)$ and $\mathbf{s}_{ii',k}(t)$ is always unique (*i.e.,* a tie never occurs). The decoder adds $\mathbf{s}_r^*(t)$ and $\mathbf{s}_k^*(t)$ to obtain $\mathbf{u}(t)$ (fed to the plant) as shown below:

$$\mathbf{u}(t) = \mathbf{s}_r^*(t) + \mathbf{s}_k^*(t). \tag{19}$$

B. Correctness

We first prove the following claim which we use in the proof of correctness.

Claim 2: Assuming $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are distinct and non-zero, and the operations of ENC_I, ENC_O, *DEC* and observers are as described in Section IV-A, the following are true (even in the presence of a 1-active adversary):

(a) For time $t \geq 0$, $\mathbf{s}_r^*(t) = \mathbf{r}(t)$.

(b) If observer i is not under the adversary's influence and $\mathbf{K}\hat{\mathbf{x}}_i(t) = \mathbf{K}\mathbf{x}(t) + \lambda_i\mathbf{K}\boldsymbol{\Delta}(t)$ holds at time t, then $\mathbf{s}_k^*(t) = \mathbf{K}\mathbf{x}(t)$.

where $\boldsymbol{\Delta}(t) \in \mathbf{R}^n$ in (b) is arbitrary.

Proof: We first describe the proof of (a) as follows. When the adversary does not inject errors in $\tilde{\mathbf{r}}_i(t)$ (*i.e.,* $\tilde{\mathbf{r}}_i(t) = \mathbf{r}_i(t) = \mathbf{r}(t) + \lambda_i\boldsymbol{\theta}(t)$), it is easy to verify that all the 6 possible $\mathbf{s}_{ii',r}(t)$ are equal to $\mathbf{r}(t)$; hence $\mathbf{s}_r^*(t) = \mathbf{r}(t)$. When there is a 1-active adversary, the majority value $\mathbf{s}_r^*(t)$ is still unique and equal to $\mathbf{r}(t)$. To check this, consider the case when a non-zero error $\mathbf{e}_{1,r}(t)$ is introduced by the adversary in $\tilde{\mathbf{r}}_1(t)$ (*i.e.,* $\tilde{\mathbf{r}}_1(t) = \mathbf{r}_1(t) + \mathbf{e}_{1,r}(t)$). In this case, it is easy to verify that due to distinct $\lambda_1, \lambda_2, \lambda_3$ and λ_4, $\mathbf{s}_{12,r}(t) \neq \mathbf{s}_{13,r}(t) \neq \mathbf{s}_{14,r}(t)$ while $\mathbf{s}_{23,r}(t) = \mathbf{s}_{34,r}(t) = \mathbf{s}_{24,r}(t)$ leads to the majority value $\mathbf{r}(t)$. Similarly, it can be easily verified for the case when the adversary attacks observer $i \in \{2,3,4\}$ that the majority value $\mathbf{s}_r^*(t)$ is unique and equal to $\mathbf{r}(t)$. The proof of (b) is similar to the proof of (a), and we skip it for brevity. ■

The following claim is sufficient to show correctness.

Claim 3: Assuming $\lambda_1, \lambda_2, \lambda_3$ and λ_4 are distinct and non-zero, and the operations of ENC_I, ENC_O, *DEC* and observers are as described in Section IV-A, the following are true for time $t \geq 0$:

(a) $\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t) + \mathbf{r}(t)$.

(b) If observer i is not under the adversary's influence, $\hat{\mathbf{x}}_i(t) = \mathbf{x}(t) + \lambda_i\boldsymbol{\Delta}(t)$. In addition, $\boldsymbol{\Delta}(t) \in \mathbf{R}^n$ satisfies the following: $\boldsymbol{\Delta}(0) = \mathbf{0}$ and $\boldsymbol{\Delta}(t+1) = (\mathbf{A} + \mathbf{B}\mathbf{K} - \mathbf{L}\mathbf{C})\boldsymbol{\Delta}(t) + \mathbf{B}\boldsymbol{\theta}(t) + \mathbf{L}\boldsymbol{\delta}(t)$.

Proof: The proof is by induction and utilizes Claim 2 (see [15] for details). ■

Since $\mathbf{u}(t) = \mathbf{K}\mathbf{x}(t) + \mathbf{r}(t)$ leads to the plant's state sequence shown in (5), the correctness constraint is satisfied.

C. Secrecy

The observations of a 1-active adversary in the 4-observer setup are similar to that of a 1-passive adversary in Section III-C, *i.e.,* observations are in the form of an affine model in the parameter $\mathbf{r}_{0:l-1}$, similar to (12)). For an adversary attacking observer i, the CRLB leads to the following bound:

$$\mathbf{E}_{r,0:l-1} \succeq \lambda_i^2\sigma^2\left(\mathbf{H}_l^T\mathbf{H}_l\right)^{-1} \tag{20}$$

where $\mathbf{E}_{r,0:l-1}$ is the error covariance matrix for adversary's MVU estimate of $\mathbf{r}_{0:l-1}$, and $\mathbf{H}_l$ is as defined in (12).

V. ρ-ACTIVE ADVERSARY

In this section, we generalize the results in Section IV from a 1-active adversary to a ρ-active adversary. This generalization is based on a class of error correcting codes called Reed-Solomon codes [19], [20], [7]; we briefly describe the idea behind this generalization in Section V-A. We then describe the proposed $3\rho+1$-observer setup (in Section V-B) and prove that it satisfies the correctness and secrecy constraints (in Section V-C) against a ρ-active adversary. As a result, $d_{min} = 3\rho + 1$ for a ρ-active adversary[1].

A. Reed-Solomon codes

Consider a polynomial $f(\lambda) = \sum_{j=0}^{\rho} c_j\lambda^j$ with coefficients $c_j \in \mathbb{R}$ and degree at most ρ. For $i \in \{1,2,\ldots w\}$ let d_i be the evaluation of f at distinct and non-zero points $\lambda_i \in \mathbb{R}$, *i.e.,* $d_i = f(\lambda_i)$. Now, consider the problem of finding c_0 from evaluations $d_1, d_2, \ldots d_w$ when any q of the evaluations are arbitrarily erroneous. It can be shown that if $q < \frac{w-\rho}{2}$, c_0 can be recovered by finding the polynomial which *fits* the maximum number of evaluations [19]. The above problem is also the same as decoding a Reed-Solomon code where c_0 is a message symbol and $d_1, d_2, \ldots d_w$ are codeword symbols [19], [20]. This observation provides an alternative interpretation of the decoder's operation in the 4-observer setup in Section IV-A. In the absence of adversarial corruptions, the decoder receives $\mathbf{r}_i(t) = \mathbf{r}(t) + \lambda_i\boldsymbol{\theta}(t)$ which is essentially a system of polynomials (of degree at most 1) evaluated at $\lambda = \lambda_i$. Hence, the task of finding $\mathbf{r}(t)$ using evaluations $\tilde{\mathbf{r}}_1(t), \tilde{\mathbf{r}}_2(t), \ldots \tilde{\mathbf{r}}_4(t)$ (with at most one erroneous evaluation) is equivalent to a decoding a Reed-Solomon code. For decoding a Reed-Solomon code, the approach of finding the best fitting polynomial still works but there exist faster methods (*e.g.,* Berlekamp-Welch algorithm [21]) whose time

[1] It can be shown that without secrecy constraints, $d_{min} = 2\rho + 1$ against a ρ-active adversary; if $2\rho + 1 < d < 3\rho + 1$, secrecy against $d - (2\rho + 1)$ compromised observers can still be guaranteed (details in [15]).

complexity is polynomial in number of evaluations. We generalize the ideas discussed above for ensuring correctness and secrecy in a $3\rho+1$-observer setup against a ρ-active adversary (by using polynomials of degree at most ρ).

B. $3\rho+1$-*observer setup*

The operations of the encoders, observers (indexed by i) and decoder are described below.

Encoders: For $i \in \{1, 2, \dots 3\rho+1\}$, the following operation is done at ENC_I which receives $\mathbf{r}(t)$ as input:

$$\mathbf{r}_i(t) = \mathbf{r}(t) + \sum_{j=1}^{\rho} \lambda_i^j \boldsymbol{\theta}_j(t) \tag{21}$$

where $\boldsymbol{\theta}_j(t)$ is a random vector $\sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}_m\right)$ generated by ENC_I and is distributed i.i.d. over time. Also, for $j \neq j'$, $\boldsymbol{\theta}_j(t)$ and $\boldsymbol{\theta}_{j'}(t)$ are independent. The scaling factor $\lambda_i \in \mathbb{R} - \{0\}$ is the same for all time t for observer i (and is assumed to be distinct across the observers *i.e.,* $\lambda_i \neq \lambda_{i'}$ where $i \neq i'$). Clearly, $\mathbf{r}_i(t)$ corresponds to the evaluation of $\mathbf{r}(t) + \sum_{j=1}^{\rho} \lambda^j \boldsymbol{\theta}_j(t)$ at $\lambda = \lambda_i$. Similarly, the following operation is done by the output encoder ENC_O:

$$\mathbf{y}_i(t) = \mathbf{y}(t) + \sum_{j=1}^{\rho} \lambda_i^j \boldsymbol{\delta}_j(t) \tag{22}$$

where $\boldsymbol{\delta}_j(t)$ is a random vector $\sim \mathcal{N}\left(\mathbf{0}, \sigma^2 \mathbf{I}_p\right)$ generated by ENC_O and is distributed i.i.d. over time. Also, for $j \neq j'$, $\boldsymbol{\delta}_j(t)$ and $\boldsymbol{\delta}_{j'}(t)$ are independent. The adversary is assumed to have knowledge of the scaling factor λ_i for each observer. The random vectors generated by ENC_I and ENC_O are assumed to be independent.

Observers: The operations done at an observer which is not under the influence of an adversary are described below. For $i \in \{1, 2, \dots 3\rho+1\}$, observer i receives $\mathbf{r}_i(t)$ and $\mathbf{y}_i(t)$ at time t and uses update rule (6) for its state estimate $\hat{\mathbf{x}}_i(t)$. Observer i has knowledge of $\mathbf{x}(0)$ and sets its initial state estimate as $\hat{\mathbf{x}}_i(0) = \mathbf{x}(0)$. A ρ-active adversary can attack any ρ observers and arbitrarily change their operations.

Decoder: For $i \in \{1, 2, \dots 3\rho+1\}$, decoder DEC receives $\tilde{\mathbf{r}}_i(t)$ and $\tilde{\mathbf{k}}_i(t)$ at time t which correspond to polynomial evaluations (of degree at most ρ) at λ_i; evaluations corresponding to attacked observers can be erroneous. DEC computes $\mathbf{r}(t)$ and $\mathbf{Kx}(t)$ using a Reed-Solomon decoder (details in [15]) and realizes plant input $\mathbf{u}(t) = \mathbf{r}(t) + \mathbf{Kx}(t)$.

C. Correctness and secrecy

The proof of correctness follows along the same lines as that for a 1-active adversary in Section IV-B. We skip the details here for brevity. For the secrecy analysis, consider the case when the adversary attacks observers $a_1, a_2, \dots a_\rho \in \{1, 2, \dots 3\rho+1\}$. The CRLB for the MVU estimator for $\mathbf{r}_{0:l-1}$ in this case is as shown below:

$$\mathbf{E}_{r,0:l-1} \succeq \frac{\sigma^2 \left(\mathbf{H}_l^T \mathbf{H}_l\right)^{-1}}{\boldsymbol{\eta} \left(\Lambda \Lambda^T\right)^{-1} \boldsymbol{\eta}^T} \tag{23}$$

where $\mathbf{E}_{r,0:l-1}$ is the error covariance matrix for the MVU estimate of $\mathbf{r}_{0:l-1}$, $\boldsymbol{\eta} = [1\ 1\ \dots 1]$, $\mathbf{H}_l$ is as defined in (12) and matrix Λ is as shown below:

$$\Lambda = \begin{bmatrix} \lambda_{a_1} & \lambda_{a_1}^2 & \dots & \lambda_{a_1}^{\rho} \\ \lambda_{a_2} & \lambda_{a_2}^2 & \dots & \lambda_{a_2}^{\rho} \\ \vdots & \vdots & \vdots & \vdots \\ \lambda_{a_\rho} & \lambda_{a_\rho}^2 & \dots & \lambda_{a_\rho}^{\rho} \end{bmatrix}. \tag{24}$$

References

[1] A. Cardenas, S. Amin, B. Sinopoli, A. Giani, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," in *Workshop on Future Directions in Cyber-Physical Systems Security*, July 2009.

[2] Y. Mo, T.-H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber-physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, Jan. 2012.

[3] NIST, "Foundations for Innovations in Cyber-Physical Systems Workshop report," Jan 2013. [Online]. Available: http://www.nist.gov/el/upload/CPS-WorkshopReport-1-30-13-Final.pdf

[4] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proceedings of the 1st ACM International Conference on High Confidence Networked Systems (HiCoNS)*, 2012, pp. 55–64.

[5] J. Villasenor, "Compromised by design? securing the defense electronics supply chain," *Brookings Institution Report*, Nov 2013. [Online]. Available: http://iis-db.stanford.edu/pubs/24484/Villasenor-Securing_the_Defense_Electronics_Supply_Chain.pdf

[6] D. Dolev, C. Dwork, O. Waarts, and M. Yung, "Perfectly secure message transmission," *Journal of the ACM*, vol. 40, no. 1, pp. 17–47, Jan. 1993.

[7] R. Blahut, *Algebraic Codes for Data Transmission.* Cambridge University Press, 2003.

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

[9] S. S. Kia, J. Cortes, and S. Martinez, "Dynamic average consensus under limited control authority and privacy requirements," *preprint*, 2014. [Online]. Available: http://arxiv.org/pdf/1401.6463v1.pdf

[10] D. Chaum, C. Crépeau, and I. Damgard, "Multiparty unconditionally secure protocols," in *Proceedings of the 20th Annual ACM Symposium on Theory of Computing (STOC)*, 1988, pp. 11–19.

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

[12] J. Le Ny and G. J. Pappas, "Differentially private Kalman filtering," in *Proceedings of the 50th Annual Allerton Conference on Communication, Control, and Computing*, 2012, pp. 1618–1625.

[13] W. A. Malik, N. C. Martins, and A. Swami, "LQ control under security constraints," in *Control of Cyber-Physical Systems.* Springer, 2013, pp. 101–120.

[14] P. Antsaklis and A. Michel, *Linear Systems.* Birkhäuser Boston, 2005.

[15] S. Mishra, N. Karamchandani, P. Tabuada, and S. Diggavi, "Secure state estimation and control using multiple (insecure) observers," *Extended version.* [Online]. Available: https://sites.google.com/site/shaunakmishracomm/home/publications/cdc14

[16] S. Kay, *Fundamentals of Statistical Signal Processing: Estimation Theory.* Prentice-Hall PTR, 1998.

[17] Y. C. Eldar, "Minimum variance in biased estimation: Bounds and asymptotically optimal estimators," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1915–1930, July 2004.

[18] S. Park, E. Serpedin, and K. Qaraqe, "Gaussian assumption: The least favorable but the most useful [lecture notes]," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 183–186, Mar. 2013.

[19] J. Wolf, "An introduction to Reed-Solomon codes," Course notes. [Online]. Available: http://pfister.ee.duke.edu/courses/ecen604/rspoly.pdf

[20] R. J. McEliece and D. V. Sarwate, "On sharing secrets and Reed-Solomon codes," *Communications of the ACM*, vol. 24, no. 9, pp. 583–584, Sept. 1981.

[21] E. Berlekamp, "Bounded distance+1 soft-decision Reed-Solomon decoding," *IEEE Transactions on Information Theory*, vol. 42, no. 3, pp. 704–720, May 1996.