2018 IEEE Conference on Control Technology and Applications (CCTA)
Copenhagen, Denmark, August 21-24, 2018

Secure Sensor Design for Resiliency of Control Systems Prior to Attack Detection*

Muhammed O. Sayin and Tamer Başar

***Abstract*— We introduce a new defense mechanism for stochastic control systems with control objectives, to enhance their resilience before the detection of any attacks. To this end, we cautiously design the outputs of the sensors that monitor the state of the system since the attackers need the sensor outputs for their malicious objectives in stochastic control scenarios. Different from the defense mechanisms that seek to detect infiltration or to improve detectability of the attacks, the proposed approach seeks to minimize the damage of possible attacks before they actually have even been detected. We, specifically, consider a controlled Gauss-Markov process, where the controller could have been infiltrated into at any time within the system's operation. Within the framework of game-theoretic hierarchical equilibrium, we provide a semi-definite programming based algorithm to compute the optimal linear secure sensor outputs that enhance the resiliency of control systems prior to attack detection.**

I. INTRODUCTION

Incorporating cyber, i.e., Internet connected, components into control systems, cyber-physical systems, e.g., process control systems and smart grid, are vulnerable against cyber attacks [1], [2]. Robust control approaches [3] against random external disturbances may not address such attacks since those attacks are target-specific with certain long term objectives. Intrusion detection systems seek to detect misbehaviors in the system to take appropriate counter actions in order to reduce the damage due to such attacks as early as possible [4]. However, advanced and persistent attackers can also seek to deceive the detection mechanisms by manipulating monitoring signals, by attacking stealthily, or by also compromising the detection mechanisms [5]. As an example, [6] analyzes false data injection attacks, where attackers can inject data to the sensor outputs and can avoid detection mechanism strategically while degrading the estimation operations of cyber-physical systems.

In this paper, we specifically analyze the attacks with certain adversarial control objectives in linear-quadratic-Gaussian (LQG) systems. Prior literature [7]–[9] has formulated such optimal stealthy control attacks. In [7], [8], the authors have formulated the optimal attacks that can remain undetected while driving the state of the system according to his/her adversarial goal by manipulating both sensor outputs and control inputs together. Recently, [9] has formulated the optimal attack strategies that can maximize the quadratic cost of a system by keeping the Kullback-Leibler distance [10] between the realized and the desired state behaviors at minimum to avoid detection and showed that injecting independent Gaussian noise with certain variance into the control input is the optimal attack. Different from [7]–[9], another recent study [11] has proposed linear encoding schemes for sensor outputs of an LQG system in order to enhance detectability of false data injection attacks. However, the coding matrix is assumed to be unknown by the attackers and the authors have proposed to mitigate such issues via time-varying coding matrices. In spite of these extensive studies, we still have significant and yet unexplored problems about how to enhance security against undetected attacks, i.e., to reduce damage due to attacks before detection.

For resiliency of control systems prior to attack detection, we seek to design the sensor outputs a-priori such that when the controller of the system is compromised by an attacker, the damage is minimized. We restrict the sensor strategies to linear functions, which leads to an LQG control problem. Otherwise, the problem entails non-classical information model and for general sensor outputs, the corresponding optimal control strategies could not be unique and could not even be expressed in closed form [12]. Before the detection, the controller of the system could have already been compromised and disregarding such a possibility and disclosing state information to the controller as if he/she has not been compromised could benefit the attacker in his/her malicious objective. We note that due to the stochastic nature of the problem, i.e., due to state noise, the attacker needs the sensor outputs to drive the system in his/her desired path effectively [12].

Furthermore, similar to the compromise of the controller, sensors of the system could also be compromised, which can cancel the effort to disclose state information cautiously with a shortcut to the state. To mitigate such issues, we consider the scenarios where the sensors do not have access to the actual state realizations and they are not controlled externally. Particularly, all the sensor output strategies are selected and fixed a-priori. Correspondingly, the attacker by infiltrating into the system could be aware of those strategies. Even though the attacker might be aware of the selected sensor output strategies, we seek to select them such that the damage before the detection is minimized. To this end, we consider a hierarchical game framework, where the sensors are the leader of the game, by announcing their strategies beforehand, and the controller, which might be adversarial or not, is the follower of the game. The sensors should anticipate the reaction of the controller, based on certain belief about the controller's type, e.g., malicious or not, and

*This research was supported by the U.S. Office of Naval Research (ONR) MURI grant N00014-16-1-2710.

The authors are with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, Urbana, IL 61801 USA. E-mail: {sayin2,basar1}@illinois.edu

978-1-5386-7698-1/18/$31.00 ©2018 IEEE

the change of type during the operation.

We have introduced secure sensor design framework in [13], but have not completely solved the problem. We have considered the controller could only be compromised at the beginning of the operation. In [5], we have extended [13] for the scenarios where the controller could be compromised during the operation. Here, we also consider that the controller can be compromised during the operation, however, different from [5], we consider the situation where the controller can have access to the previous control inputs. For the system, this new model removes the uncertainty about how the state has been driven by an attacker before the detection. Correspondingly, for an attacker, this new model removes the uncertainty before the infiltration since an attacker could also infiltrate into a system, which has already been compromised by another attacker. Furthermore, recording the control inputs at the controller can also play a role for the forensic analysis of the attack in order to identify the attacker objective after detection.

The paper is organized as follows: In Section II, we describe the secure sensor design problem. In Section III, we characterize the optimal controller response strategies for given sensor strategies and for any type. We compute the optimal secure sensor strategies in Section IV. We conclude the paper in Section V with several remarks and possible research directions.

Notations: For an ordered set of parameters, e.g., $x_1,\cdots,x_n$, we define $x_{[k,l]} := x_k,\cdots,x_l$, where $1 \leq k \leq l \leq n$. $\mathbb{N}(0,.)$ denotes the multivariate Gaussian distribution with zero mean and designated covariance. We denote random variables by bold lower case letters, e.g., $\boldsymbol{x}$. For a random variable $\boldsymbol{x}$, $\hat{\boldsymbol{x}}$ is another random variable corresponding to its posterior belief conditioned on certain other random variables that will be apparent from the context. For a vector x and a matrix A, x' and A' denote their transposes, and $\|x\|$ denotes the Euclidean (L^2) norm of the vector x. For a matrix A, $\mathrm{tr}\{A\}$ denotes its trace. We denote the identity and zero matrices with the associated dimensions by I and O, respectively, while $\mathbf{1}$ (or $\mathbf{0}$) denotes a vector whose entries are all 1 (or 0). For positive semi-definite matrices A and B, $A \succeq B$ means that $A-B$ is also a positive semi-definite matrix. $A \otimes B$ denotes the Kronecker product of the matrices A and B.

II. Problem Formulation

Consider a controlled stochastic system described by the following equations:

$$\boldsymbol{x}_{k+1} = A\boldsymbol{x}_k + B\boldsymbol{u}_k + \boldsymbol{v}_k, \tag{1}$$

for $k = 1,2,\dots,n$, where[1] $A \in \mathbb{R}^{m\times m}$, $B \in \mathbb{R}^{m\times r}$, and $\boldsymbol{x}_k \sim \mathbb{N}(0,\Sigma_k)$, $k = 1,\dots,n$. The additive state noise sequence $\{\boldsymbol{v}_k\}$ is white Gaussian vector process, i.e., $\boldsymbol{v}_k \sim \mathbb{N}(0,\Sigma_v)$; and is independent of the initial state $\boldsymbol{x}_1$. We assume that the matrix A is non-singular, and the auto-covariance matrices Σ_1 and Σ_v are positive definite. The closed loop control vector $\boldsymbol{u}_k \in \mathbb{R}^r$ is given by

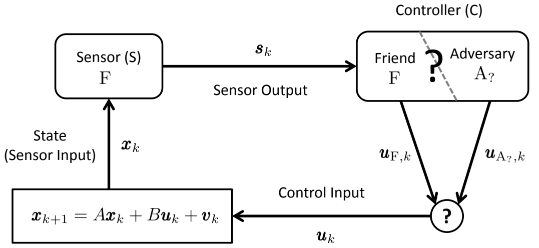

Fig. 1: Cyber physical system including a sensor and a controller.

$$\boldsymbol{u}_k = \gamma_k(\boldsymbol{s}_{[1,k]},\boldsymbol{u}_{[1,k-1]}), \tag{2}$$

where $\gamma_k(\cdot)$ can be any Borel measurable function from $\mathbb{R}^{mk+r(k-1)}$ to $\mathbb{R}^r$. The sensor output $\boldsymbol{s}_k \in \mathbb{R}^m$ is given by

$$\boldsymbol{s}_k = \eta_k(\boldsymbol{x}_k), \tag{3}$$

where $\eta_k(\cdot)$ can be any *linear* function from $\mathbb{R}^m$ to $\mathbb{R}^m$.

We have two separate agents: Sensor (S) and Controller (C), as seen in Fig. 1. At each stage $k = 1,\dots,n$, the agents construct $\boldsymbol{s}_k$ and $\boldsymbol{u}_k$ according to their own objectives. In particular, S chooses η_k from the strategy space Υ, which, for each k, is the set of all linear functions from $\mathbb{R}^m$ to $\mathbb{R}^m$, i.e., $\eta_k \in \Upsilon$ and $\boldsymbol{s}_k = \eta_k(\boldsymbol{x}_k)$. This implies that for each $\eta_k \in \Upsilon$, there exists a matrix $\mathscr{L}_k \in \mathbb{R}^{m\times m}$ such that

$$\boldsymbol{s}_k = \mathscr{L}_k'\boldsymbol{x}_k, \tag{4}$$

almost surely on $\mathbb{R}^m$. C chooses γ_k from the strategy space Γ_k, which is the set of all Borel measurable functions from $\mathbb{R}^{mk+r(k-1)}$ to $\mathbb{R}^r$, i.e., $\gamma_k \in \Gamma_k$ and $\boldsymbol{u}_k = \gamma_k(\boldsymbol{s}_{[1,k]})$.

While S has a single type denoted by F, C can have one of $\Theta := \{\mathrm{F},\mathrm{A}_1,\dots,\mathrm{A}_t\}$, which can also change in time. Furthermore, S does not know C's exact type. Particularly, the types $\{\mathrm{A}_1,\dots,\mathrm{A}_t\}$ correspond to those attackers that seek to infiltrate into C. Once an attacker achieves to infiltrate, he/she becomes in charge of C and can construct $\boldsymbol{u}_k$'s accordingly while S is unaware of the infiltration. Furthermore, within time, infiltration attacks can succeed or fail, and defense mechanisms can detect the attacks or not, which implies that the type of C can change dynamically. To model the type changes explicitly, we can consider a jump process $\{\boldsymbol{\theta}_j \in \Theta\}$ and we consider the scenarios where the type changes can occur at certain instances, e.g., $\Lambda \subset \{1,\dots,n\}$.

Having a single time-invariant type F, S has a single cost function:

$$J_{\mathrm{F}}(\eta_{[1,n]};\gamma_{[1,n]}) = \mathbb{E}\left\{\sum_{k=1}^{n} \|\boldsymbol{x}_{k+1}\|_{Q_{\mathrm{F}}}^2 + \|\boldsymbol{u}_k\|_{R_{\mathrm{F}}}^2\right\}, \tag{5}$$

where[2] $Q_{\mathrm{F}} \in \mathbb{R}^{m\times m}$ is positive semi-definite and $R_{\mathrm{F}} \in \mathbb{R}^{r\times r}$ is positive definite. However, based on his/her type and the

[1] Even though we consider time invariant matrices A and B for notational simplicity, the provided results could be extended to time-variant cases rather routinely. Furthermore, we consider all the random parameters to have zero mean; however, the derivations can be extended to non-zero mean case in a straight-forward way.

[2] For notational simplicity, we consider time-invariant Q_{F} and R_{F}. However, the provided results could be extended to time-variant cases rather routinely.

time the type has changed, C can have different objectives. In particular, if C has type F and the type has changed at $k=\kappa$, C's cost function is given by

$$J_{\mathrm{F}}(\eta_{[1,n]};\cdot,\gamma_{\mathrm{F},[\kappa,n]})=\mathbb{E}\left\{\sum_{k=\kappa}^{n}\|\boldsymbol{x}_{\mathrm{F},k+1}\|_{Q_{\mathrm{F}}}^{2}+\|\boldsymbol{u}_{\mathrm{F},k}\|_{R_{\mathrm{F}}}^{2}\right\}; \quad (6)$$

where '$\cdot$' as an argument of the cost function refers to C's strategies $\gamma_1,\dots,\gamma_{\kappa-1}$, which are selected by C before C has become type F, and the subscript F in the state $\boldsymbol{x}_{\mathrm{F},k}$, the strategy $\gamma_{\mathrm{F},k}$, and the control input $\boldsymbol{u}_{\mathrm{F},k}$ show their dependence on C's type explicitly. Furthermore, if C has one of the types $\{\mathrm{A}_1,\dots,\mathrm{A}_t\}$, i.e., if an attacker is in charge of C, C's cost function is given by

$$\begin{aligned}J_{\mathrm{A}_i}(\eta_{[1,n]};\cdot,\gamma_{\mathrm{A}_i,[\kappa,n]})=\mathbb{E}\Bigg\{&\sum_{k=\kappa}^{n}\|\boldsymbol{x}_{\mathrm{A}_i,k+1}-z_i\|_{Q_{\mathrm{A}_i}}^{2}\\&+\|\boldsymbol{u}_{\mathrm{A}_i,k}-\boldsymbol{u}_{\mathrm{F},k}\|_{R_{\mathrm{A}_i}}^{2}\Bigg\}, \quad (7)\end{aligned}$$

where $Q_{\mathrm{A}_i}\in\mathbb{R}^{m\times m}$ is positive semi-definite and $R_{\mathrm{A}_i}\in\mathbb{R}^{r\times r}$ is positive definite. We note that $\boldsymbol{x}_{\mathrm{A}_i,k}$ denotes the state driven by the adversarial control input $\boldsymbol{u}_{\mathrm{A}_i,k}$ while $\boldsymbol{u}_{\mathrm{F},k}$ denotes the control input that would have been used if C would have type F so that, by being close to the desired control input, the attacker A_i can avoid intrusion detection mechanisms [13].

The agents S and C aim to minimize their cost functions by choosing the strategies $\eta_{[1,n]}$ and $\gamma_{[1,n]}$ while each strategy implicitly depends on the other. Due to the hierarchy, C's strategies $\gamma_{\theta,k}\in\Gamma_k$, $\theta\in\{\mathrm{F},\mathrm{A}_1,\dots,\mathrm{A}_t\}$, depending on his/her type, can also depend on S's strategies $\eta_{[1,k]}$ and the time of type change. In order to show these dependences explicitly, henceforth, we denote C's strategies by $\gamma_{\theta,k}^{(\kappa)}(\eta_{[1,k]})$, which implies $\gamma_{\theta,k}^{(\kappa)}(\eta_{[1,k]})(\boldsymbol{s}_{[1,k]}):=\gamma_{\theta,k}(\boldsymbol{s}_{[1,k]})$. Then, the pair of strategies:

$$\left[\eta_{[1,n]}^{*};\left(\gamma_{\theta,[\kappa,n]}^{(\kappa)*},\theta\in\Theta,\kappa\in\hbar\right)\right] \quad (8)$$

attains the Stackelberg equilibrium provided that

$$\eta_{[1,n]}^{*}=\underset{\substack{\eta_k\in\Upsilon,\\ k=1,\dots,n}}{\operatorname{argmin}}\ \mathbb{E}\left\{\sum_{\kappa,\kappa_+\in\Lambda}\sum_{k=\kappa}^{\kappa_+-1}\|\boldsymbol{x}_{\boldsymbol{\theta}_j,k+1}\|_{Q_{\mathrm{F}}}^{2}+\|\boldsymbol{u}_{\boldsymbol{\theta}_j,k}\|_{R_{\mathrm{F}}}^{2}\right\} \quad (9a)$$

$$\gamma_{\theta,[\kappa,n]}^{(\kappa)*}(\eta_{[1,n]})=\underset{\substack{\gamma_{\theta,k}^{(\kappa)}\in\Gamma_k,\\ k=\kappa,\dots,n}}{\operatorname{argmin}}\ J_{\theta,\kappa}\left(\eta_{[1,n]};\cdot,\gamma_{\theta,[\kappa,n]}^{(\kappa)}(\eta_{[1,n]})\right), \quad (9b)$$

where the expectation is also taken over $\{\boldsymbol{\theta}_j\}$, κ_+ is the type change time after κ in Λ, and $\boldsymbol{\theta}_j$ refers to the type of C when $k\in[\kappa,\kappa_+)$. We note that the optimization in (9b) results in an equivalence class of strategies such that all optimizing strategies lead to the same control input $\boldsymbol{u}_{\boldsymbol{\theta},k}$ almost everywhere on $\mathbb{R}^r$ [5].

III. OPTIMAL C STRATEGIES FOR ANY TYPE

For given linear S strategies, optimal C strategies can be computed as in [5], however, here, C can also have access to the previous control inputs. Since C can have access to the previous control inputs, C does not need to know what C's type was. Correspondingly, we can relax the assumption that attackers consider C's type was F before the type change, i.e., before the infiltration.

Based on[3] [5], when C has type F, the optimal control inputs $\boldsymbol{u}_{\mathrm{F},[\kappa,n]}$ are given by

$$\begin{bmatrix}\boldsymbol{u}_{\mathrm{F},n}^{*}\\ \vdots\\ \boldsymbol{u}_{\mathrm{F},\kappa}^{*}\end{bmatrix}=-\left(\Phi_{\mathrm{F}}^{(\kappa)}\right)^{-1}\left(K_{\mathrm{F}}^{(\kappa)}\begin{bmatrix}\hat{\boldsymbol{x}}_{n}^{o}\\ \vdots\\ \hat{\boldsymbol{x}}_{\kappa}^{o}\end{bmatrix}+\underline{\Phi}_{\mathrm{F}}^{(\kappa)}\begin{bmatrix}\boldsymbol{u}_{\kappa-1}\\ \vdots\\ \boldsymbol{u}_{1}\end{bmatrix}\right), \quad (10)$$

where $\hat{\boldsymbol{x}}_k^o=\mathbb{E}\{\boldsymbol{x}_k^o|\boldsymbol{s}_{[1,k]}\}$, the control-free state $\boldsymbol{x}_k^o$ evolves according to

$$\boldsymbol{x}_{k+1}^{o}=A\boldsymbol{x}_{k}^{o}+\boldsymbol{v}_{k}, \quad (11)$$

and the matrices $\Phi_{\mathrm{F}}^{(\kappa)}\in\mathbb{R}^{(n-\kappa+1)r\times(n-\kappa+1)r}$, $K_{\mathrm{F}}^{(\kappa)}\in\mathbb{R}^{(n-\kappa+1)r\times(n-\kappa+1)m}$, $\underline{\Phi}_{\mathrm{F}}^{(\kappa)}\in\mathbb{R}^{(n-\kappa+1)r\times(\kappa-1)r}$ are defined by

$$\Phi_{\mathrm{F}}^{(\kappa)}:=\begin{bmatrix}I & K_{\mathrm{F},n}B & \cdots & K_{\mathrm{F},n}A^{n-\kappa-1}B\\ & I & \cdots & K_{\mathrm{F},n-1}A^{n-\kappa-2}B\\ & & \ddots & \vdots\\ & & & I\end{bmatrix},\ K_{\mathrm{F}}^{(\kappa)}:=\begin{bmatrix}K_{\mathrm{F},n} & & \\ & \ddots & \\ & & K_{\mathrm{F},\kappa}\end{bmatrix},$$

$$\underline{\Phi}_{\mathrm{F}}^{(\kappa)}:=\begin{bmatrix}K_{\mathrm{F},n}A^{n-\kappa}B & \cdots & K_{\mathrm{F},n}A^{n-2}B\\ K_{\mathrm{F},n-1}A^{n-\kappa-1}B & \cdots & K_{\mathrm{F},n-1}A^{n-3}B\\ \vdots & & \vdots\\ K_{\mathrm{F},\kappa}B & \cdots & K_{\mathrm{F},\kappa}A^{\kappa-2}B\end{bmatrix} \quad (12)$$

while

$$\Delta_{\mathrm{F},k}:=B'\check{Q}_{\mathrm{F},k+1}B+R_{\mathrm{F}},\ K_{\mathrm{F},k}:=\Delta_{\mathrm{F},k}^{-1}B'\check{Q}_{\mathrm{F},k+1}A, \quad (13)$$

$$\begin{aligned}&\check{Q}_{\mathrm{F},k}=Q_{\mathrm{F}}+A'(\check{Q}_{\mathrm{F},k+1}-\check{Q}_{\mathrm{F},k+1}B\Delta_{\mathrm{F},k}^{-1}B'\check{Q}_{\mathrm{F},k+1})A,\\&\check{Q}_{\mathrm{F},n+1}=Q_{\mathrm{F}}. \quad (14)\end{aligned}$$

Furthermore, when C has type A_i, we let

$$\bar{A}_k:=\left[\begin{array}{c:c}A & O_{m\times(n-k)r}\quad B\quad O_{m\times((k-1)r+m)}\\ \hdashline O_{(m+nr)\times m} & I_{m+nr}\end{array}\right],$$

$$\bar{B}:=\left[\begin{array}{c}B\\ \hdashline O_{(m+nr)\times r}\end{array}\right],\ \bar{Q}_{\mathrm{A}_i}:=\begin{bmatrix}I_m & O_{m\times nr} & -I_m\end{bmatrix}Q_{\mathrm{A}_i}\begin{bmatrix}I_m\\ O_{nr\times m}\\ -I_m\end{bmatrix},$$

and we introduce $\bar{\boldsymbol{x}}_{\mathrm{A}_i,k}^{o}$, evolving according to[4]

$$\underbrace{\left[\begin{array}{c}\check{\boldsymbol{x}}_{k+1}\\ \hdashline \boldsymbol{u}_{\mathrm{F}}\\ z_i\end{array}\right]}_{=:\,\bar{\boldsymbol{x}}_{k+1}^{o}}=\bar{A}_k\left[\begin{array}{c}\check{\boldsymbol{x}}_{k}\\ \hdashline \boldsymbol{u}_{\mathrm{F}}\\ z_i\end{array}\right]+\left[\begin{array}{c}I_m\\ \hdashline O_{(nr+m)\times m}\end{array}\right]\boldsymbol{v}_k,$$

where $\check{\boldsymbol{x}}_k$ is the state that would have been realized if C only has type F. Then, the optimal control input is given by

$$\begin{aligned}\begin{bmatrix}\boldsymbol{u}_{\mathrm{A}_i,n}^{*}\\ \vdots\\ \boldsymbol{u}_{\mathrm{A}_i,\kappa}^{*}\end{bmatrix}=\begin{bmatrix}\boldsymbol{u}_{\mathrm{F},n}^{*}\\ \vdots\\ \boldsymbol{u}_{\mathrm{F},\kappa}^{*}\end{bmatrix}-\left(\Phi_{\mathrm{A}_i}^{(\kappa)}\right)^{-1}\Bigg(&K_{\mathrm{A}_i}^{(\kappa)}\begin{bmatrix}\mathbb{E}\{\bar{\boldsymbol{x}}_{\mathrm{A}_i,n}^{o}|\boldsymbol{s}_{[1,n]}\}\\ \vdots\\ \mathbb{E}\{\bar{\boldsymbol{x}}_{\mathrm{A}_i,\kappa}^{o}|\boldsymbol{s}_{[1,\kappa]}\}\end{bmatrix}\\&+\underline{\Phi}_{\mathrm{A}_i}^{(\kappa)}\begin{bmatrix}\boldsymbol{u}_{\kappa-1}-\boldsymbol{u}_{\mathrm{F},\kappa-1}^{*}\\ \vdots\\ \boldsymbol{u}_{1}-\boldsymbol{u}_{\mathrm{F},1}^{*}\end{bmatrix}\Bigg), \quad (15)\end{aligned}$$

[3] Detailed derivations could be found in [5].

[4] $\bar{\boldsymbol{x}}_{\mathrm{A}_i,k}^{o}$ depends on type A_i due to z_i.

where the matrices $\Phi_{\mathrm{A}_i}^{(\kappa)} \in \mathbb{R}^{(n-\kappa+1)r\times(n-\kappa+1)r}$, $K_{\mathrm{A}_i}^{(\kappa)} \in \mathbb{R}^{(n-\kappa+1)r\times(n-\kappa+1)m}$, $\underline{\Phi}_{\mathrm{A}_i}^{(\kappa)} \in \mathbb{R}^{(n-\kappa+1)r\times(\kappa-1)r}$ are defined by

$$\Phi_{\mathrm{A}_i}^{(\kappa)} := \begin{bmatrix} I \; K_{\mathrm{A}_i,n}\bar{B} & \cdots & K_{\mathrm{A}_i,n}\bar{A}_{n-1}\ldots\bar{A}_{\kappa+1}\bar{B} \\ I & \cdots & K_{\mathrm{A}_i,n-1}\bar{A}_{n-2}\ldots\bar{A}_{\kappa+1}\bar{B} \\ & \ddots & \vdots \\ & & I \end{bmatrix}, \tag{16}$$

$$\underline{\Phi}_{\mathrm{A}_i}^{(\kappa)} := \begin{bmatrix} K_{\mathrm{A}_i,n}\bar{A}_{n-1}\ldots\bar{A}_{\kappa}\bar{B} & \cdots & K_{\mathrm{A}_i,n}\bar{A}_{n-1}\ldots\bar{A}_{2}\bar{B} \\ K_{\mathrm{A}_i,n-1}\bar{A}_{n-2}\ldots\bar{A}_{\kappa}\bar{B} & \cdots & K_{\mathrm{A}_i,n-1}\bar{A}_{n-2}\ldots\bar{A}_{2}\bar{B} \\ \vdots & & \vdots \\ K_{\mathrm{A}_i,\kappa}\bar{B} & \cdots & K_{\mathrm{A}_i,\kappa}\bar{A}_{\kappa-1}\ldots\bar{A}_{2}\bar{B} \end{bmatrix},$$

$$K_{\mathrm{A}_i}^{(\kappa)} := \begin{bmatrix} K_{\mathrm{A}_i,n} & & \\ & \ddots & \\ & & K_{\mathrm{A}_i,\kappa} \end{bmatrix},$$

while

$$\Delta_{\mathrm{A}_i,k} := \bar{B}'\check{Q}_{\mathrm{A}_i,k+1}\bar{B} + R_{\mathrm{A}_i},\; K_{\mathrm{A}_i,k} := \Delta_{\mathrm{A}_i,k}^{-1}\bar{B}'\check{Q}_{\mathrm{A}_i,k+1}\bar{A}_k, \tag{17}$$
$$\check{Q}_{\mathrm{A}_i,k} = \bar{Q}_{\mathrm{A}_i} + \bar{A}_k'(\check{Q}_{\mathrm{A}_i,k+1} - \check{Q}_{\mathrm{A}_i,k+1}\bar{B}\Delta_{\mathrm{A}_i,k}^{-1}\bar{B}'\check{Q}_{\mathrm{A}_i,k+1})\bar{A}_k,$$
$$\check{Q}_{\mathrm{A}_i,n+1} = \bar{Q}_{\mathrm{A}_i}.$$

In (15), the conditional expectation $\mathbb{E}\{\bar{\boldsymbol{x}}_{\mathrm{A}_i,k}^{o}|\boldsymbol{s}_{[1,k]}\}$ is given by

$$\mathbb{E}\{\bar{\boldsymbol{x}}_{\mathrm{A}_i,k}^{o}|\boldsymbol{s}_{[1,k]}\} = \underbrace{\begin{bmatrix} E_k - \Psi_k\Phi_{\mathrm{F}}^{-1}K_{\mathrm{F}}L_k \\ -\Phi_{\mathrm{F}}^{-1}K_{\mathrm{F}}L_k \\ O_m \end{bmatrix}}_{=:\,F_k} \underbrace{\begin{bmatrix} \hat{\boldsymbol{x}}_n^{o} \\ \vdots \\ \hat{\boldsymbol{x}}_1^{o} \end{bmatrix}}_{=:\,\hat{\boldsymbol{x}}^{o}} + \underbrace{\begin{bmatrix} O_{m\times 1} \\ O_{nr\times 1} \\ z_i \end{bmatrix}}_{=:\,\underline{z}_i}, \tag{18}$$

where $E_k := \left[O_{m\times(n-k)m}\; I_m\; O_{m\times(k-1)m}\right]$ is the indicator matrix such that $\mathbb{E}\{\boldsymbol{x}_k^{o}|\boldsymbol{s}_{[1,k]}\} = E_k\hat{\boldsymbol{x}}^{o}$, $k = 1,\ldots,n$, and

$$\Psi_k := \begin{bmatrix} O_{m\times(n-k+1)m} & B & AB & \cdots & A^{k-2}B \end{bmatrix},$$

$$L_k := \left[\begin{array}{c:c:c} & A^{n-k} & \\ & \vdots & \\ O & & O \\ & A & \\ & I_m & \\ \hdashline O & O & I_{(k-1)m} \end{array}\right].$$

Then, the optimal control inputs $\boldsymbol{u}_{\mathrm{A}_i,[\kappa,n]}$ are given by

$$\begin{bmatrix} \boldsymbol{u}_{\mathrm{A}_i,n}^{*} \\ \vdots \\ \boldsymbol{u}_{\mathrm{A}_i,\kappa}^{*} \end{bmatrix} = \begin{bmatrix} \boldsymbol{u}_{\mathrm{F},n}^{*} \\ \vdots \\ \boldsymbol{u}_{\mathrm{F},\kappa}^{*} \end{bmatrix} - \left(\Phi_{\mathrm{A}_i}^{(\kappa)}\right)^{-1}\left(K_{\mathrm{A}_i}^{(\kappa)}(F^{(\kappa)}\hat{\boldsymbol{x}}^{o} + \mathbf{1}_{n-\kappa+1}\otimes\underline{z}_i)\right.$$
$$\left. + \underline{\Phi}_{\mathrm{A}_i}^{(\kappa)}\begin{bmatrix} \boldsymbol{u}_{\kappa-1} - \boldsymbol{u}_{\mathrm{F},\kappa-1}^{*} \\ \vdots \\ \boldsymbol{u}_{1} - \boldsymbol{u}_{\mathrm{F},1}^{*} \end{bmatrix}\right), \tag{19}$$

where $F^{(\kappa)} := \left[F_n' \cdots F_\kappa'\right]'$.

In the next section, we seek to compute optimal S strategies based on (10) and (19).

IV. OPTIMAL S STRATEGIES

Note that the optimal control varies according to the C's type and the time of type change. Therefore, we first seek to write the optimal control in a unified compact form. To this end, let

$$\boldsymbol{u}_{\boldsymbol{\theta}}^{(\kappa)*} := \begin{bmatrix} \boldsymbol{u}_{\boldsymbol{\theta},n}^{*} \\ \vdots \\ \boldsymbol{u}_{\boldsymbol{\theta},\kappa}^{*} \end{bmatrix} \text{ and } \bar{\boldsymbol{u}}^{(\kappa)} := \begin{bmatrix} \boldsymbol{u}_{\kappa-1} \\ \vdots \\ \boldsymbol{u}_{1} \end{bmatrix}. \tag{20}$$

Then, by (10), $\boldsymbol{u}_{\mathrm{F}}^{(\kappa)}$ can be written as

$$\boldsymbol{u}_{\mathrm{F}}^{(\kappa)*} = -\overbrace{\left((\Phi_{\mathrm{F}}^{(\kappa)})^{-1}K_{\mathrm{F}}^{(\kappa)}\left[I_{(n-\kappa+1)m}\; O_{(n-\kappa+1)m\times(\kappa-1)m}\right]\right)}^{=:\,T_{\mathrm{F}}^{(\kappa)}}\hat{\boldsymbol{x}}^{o}$$
$$-\underbrace{\left((\Phi_{\mathrm{F}}^{(\kappa)})^{-1}\underline{\Phi}_{\mathrm{F}}^{(\kappa)}\right)}_{=:\,\underline{T}_{\mathrm{F}}^{(\kappa)}}\bar{\boldsymbol{u}}^{(\kappa)}. \tag{21}$$

Correspondingly, by (19), $\boldsymbol{u}_{\mathrm{A}_i}^{(\kappa)}$ can be written as

$$\boldsymbol{u}_{\mathrm{A}_i}^{(\kappa)*} = -(\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}K_{\mathrm{A}_i}^{(\kappa)}F^{(\kappa)}\hat{\boldsymbol{x}}^{o} - (\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}K_{\mathrm{A}_i}^{(\kappa)}\mathbf{1}_{n-\kappa+1}\otimes\underline{z}_i$$
$$-(\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}\underline{\Phi}_{\mathrm{A}_i}^{(\kappa)}bu^{(\kappa)} + \left[I_{n-\kappa+1}\; (\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}\underline{\Phi}_{\mathrm{A}_i}^{(\kappa)}\right]\boldsymbol{u}_{\mathrm{F}}^{(1)*},$$

and by (21), we obtain

$$\boldsymbol{u}_{\mathrm{A}_i}^{(\kappa)*} = -\overbrace{\left((\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}\underline{\Phi}_{\mathrm{A}_i}^{(\kappa)}\right)}^{=:\,\underline{T}_{\mathrm{A}_i}^{(\kappa)}}\bar{\boldsymbol{u}}^{(\kappa)} - \overbrace{(\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}K_{\mathrm{A}_i}^{(\kappa)}\mathbf{1}\otimes\underline{z}_i}^{=:\,Z_{\mathrm{A}_i}^{(\kappa)}}$$
$$-\underbrace{\left((\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}K_{\mathrm{A}_i}^{(\kappa)}F^{(\kappa)} + \left[I\; (\Phi_{\mathrm{A}_i}^{(\kappa)})^{-1}\underline{\Phi}_{\mathrm{A}_i}^{(\kappa)}\right](\Phi_{\mathrm{F}}^{(1)})^{-1}K_{\mathrm{F}}^{(1)}\right)}_{=:\,T_{\mathrm{A}_i}^{(\kappa)}}\hat{\boldsymbol{x}}^{o}.$$

Let $Z_{\mathrm{F}}^{(\kappa)} = 0$ be a zero vector; then for $\theta \in \Theta$, we obtain a compact form representation for the optimal control:

$$\boldsymbol{u}_{\theta}^{(\kappa)} = -T_{\theta}^{(\kappa)}\hat{\boldsymbol{x}}^{o} - \underline{T}_{\theta}^{(\kappa)}\bar{\boldsymbol{u}}^{(\kappa)} - Z_{\theta}^{(\kappa)}. \tag{22}$$

Note also that in (9a), only $\boldsymbol{u}_{\boldsymbol{\theta}_j,k}^{(\kappa)*}$ for $k = \kappa,\ldots,\kappa_+ - 1$ are included. Let $N := |\Lambda|$; then, by (22), for a given realization of the process $\{\boldsymbol{\theta}_j\}$, e.g., $\theta_{[1,N]}$, we have

$$\overbrace{\left[\begin{array}{c} M_{\kappa_N}\boldsymbol{u}_{\theta_{\kappa_N}}^{(\kappa_N)*} \\ \hdashline \vdots \\ \hdashline M_{\kappa}\boldsymbol{u}_{\theta_{\kappa}}^{(\kappa)*} \\ \hdashline \vdots \\ \hdashline M_{1}\boldsymbol{u}_{\theta_{1}}^{(1)*} \end{array}\right]}^{=:\,\boldsymbol{u}_{\theta_{[1,N]}}^{*}} = -\overbrace{\left[\begin{array}{c:c:c:c:c} O & \multicolumn{4}{c}{M_{\kappa_N}\underline{T}_{\theta_{\kappa_N}}^{(\kappa_N)}} \\ \hdashline \vdots & \ddots & & & \\ \hdashline O & \cdots & O & \multicolumn{2}{c}{M_{\kappa}\underline{T}_{\theta_{\kappa}}^{(\kappa)}} \\ \hdashline \vdots & & \vdots & \ddots & \\ \hdashline O & \cdots & O & \cdots & O \end{array}\right]}^{=:\,T_{\theta_{[1,N]}}}\boldsymbol{u}_{\theta_{[1,N]}}^{*}$$
$$-\underbrace{\left[\begin{array}{c} M_{\kappa_N}T_{\theta_{\kappa_N}}^{(\kappa_N)} \\ \hdashline \vdots \\ \hdashline M_{\kappa}T_{\theta_{\kappa}}^{(\kappa)} \\ \hdashline \vdots \\ \hdashline M_{1}T_{\theta_{1}}^{(1)} \end{array}\right]}_{=:\,M_{\theta_{[1,N]}}}\hat{\boldsymbol{x}}^{o} - \underbrace{\left[\begin{array}{c} M_{\kappa_N}Z_{\theta_{\kappa_N}}^{(\kappa_N)} \\ \hdashline \vdots \\ \hdashline M_{\kappa}Z_{\theta_{\kappa}}^{(\kappa)} \\ \hdashline \vdots \\ \hdashline M_{1}Z_{\theta_{1}}^{(1)} \end{array}\right]}_{=:\,Z_{\theta_{[1,N]}}}, \tag{23}$$

where $M_\kappa \in \mathbb{R}^{(\kappa_+-\kappa)r\times(n-\kappa+1)r}$ is given by

$$M_\kappa := \left[O_{(\kappa_+-\kappa)r\times(n-\kappa_++1)r}\; I_{(\kappa_+-\kappa)r}\right]$$

and κ_N is the last state transition time. Note that in (23), $T_{\theta_{[1,N]}}$ is an upper triangular matrix, whose diagonal entries are zero, which implies that $I+T_{\theta_{[1,N]}}$ is an invertible upper triangular matrix. Therefore, by (23), we obtain

$$\boldsymbol{u}^*_{\theta_{[1,N]}} = -(I+T_{\theta_{[1,N]}})^{-1}(M_{\theta_{[1,N]}}\hat{\boldsymbol{x}}^o + Z_{\theta_{[1,N]}}). \tag{24}$$

Even though S constructs a single set of strategies $\{\eta_k \in \Upsilon\}$ without knowing C's type, the resulting sensor outputs $\{\boldsymbol{s}_k = \eta_k(\boldsymbol{x}_k)\}$ may depend on the state $\boldsymbol{x}_k$, hence C's type and correspondingly $\theta_{[1,N]}$. However, since the problem entails classical information as shown in Section IV of [5], $\hat{\boldsymbol{x}}^o$ does not depend on $\theta_{[1,N]}$. Therefore, let $\boldsymbol{u}^*_{\theta_{[1,N]},k}$ be the corresponding control input at time k according to (24) for a given realization $\theta_{[1,N]}$. Then, the objective function (9a) is given by

$$\min_{\substack{\eta_k\in\Upsilon,\\ k=1,\ldots,n}} \mathbb{E}\left\{\sum_{k=1}^{n} \|\boldsymbol{x}_{\boldsymbol{\theta}_{[1,N]},k+1}\|^2_{Q_{\mathrm{F}}} + \|\boldsymbol{u}^*_{\boldsymbol{\theta}_{[1,N]},k}\|^2_{R_{\mathrm{F}}}\right\}. \tag{25}$$

After some algebra[5], (25) can be written as

$$\min_{\substack{\eta_k\in\Upsilon,\\ k=1,\ldots,n}} \mathbb{E}\|\Phi^{(1)}_{\mathrm{F}}\boldsymbol{u}^*_{\boldsymbol{\theta}_{[1,N]}} + K^{(1)}_{\mathrm{F}}\boldsymbol{x}^o\|^2_{\Delta} + G, \tag{26}$$

where $G := \mathrm{tr}\{\Sigma_1(\check{Q}_{\mathrm{F},1} - Q_{\mathrm{F}})\} + \sum_{k=1}^{n}\mathrm{tr}\{\Sigma_v\check{Q}_{\mathrm{F},k+1}\}$,

$$\Delta := \begin{bmatrix} \Delta_{\mathrm{F},n} & & \\ & \ddots & \\ & & \Delta_{\mathrm{F},1}\end{bmatrix},$$

and $\check{Q}_{\mathrm{F},k}$ and $\Delta_{\mathrm{F},k}$ are defined in (14) and (13), respectively.

Next, we introduce the parameters:

$$\Xi_{\boldsymbol{\theta}_{[1,N]}} := -\Phi^{(1)}_{\mathrm{F}}(I+T_{\boldsymbol{\theta}_{[1,N]}})^{-1}M_{\boldsymbol{\theta}_{[1,N]}}, \tag{27a}$$

$$\xi_{\boldsymbol{\theta}_{[1,N]}} := -\Phi^{(1)}_{\mathrm{F}}(I+T_{\boldsymbol{\theta}_{[1,N]}})^{-1}Z_{\boldsymbol{\theta}_{[1,N]}}, \tag{27b}$$

almost everywhere on $\mathbb{R}^{nr\times nm}$ and $\mathbb{R}^{nm}$, respectively. Then, we obtain

$$\min_{\substack{\eta_k\in\Upsilon,\\ k=1,\ldots,n}} \mathbb{E}\|\Xi_{\boldsymbol{\theta}_{[1,N]}}\hat{\boldsymbol{x}}^o + \xi_{\boldsymbol{\theta}_{[1,N]}} + K^{(1)}_{\mathrm{F}}\boldsymbol{x}^o\|^2_{\Delta} + G, \tag{28}$$

which has identical form with equation (61) in [5]. For notational simplicity, let $K := K^{(1)}_{\mathrm{F}}$, $\Xi_{\boldsymbol{\theta}} := \Xi_{\boldsymbol{\theta}_{[1,N]}}$ and $\xi_{\boldsymbol{\theta}} := \xi_{\boldsymbol{\theta}_{[1,N]}}$. And following similar lines with [5], the optimization problem (28) can be written as

$$\min_{\substack{\eta_k\in\Upsilon,\\ k=1,\ldots,n}} \mathrm{tr}\left\{\begin{bmatrix} H_n & AH_{n-1} & \cdots & A^{n-1}H_1 \\ H_{n-1}A' & H_{n-1} & \cdots & A^{n-2}H_1 \\ \vdots & \vdots & \ddots & \vdots \\ H_1(A^{n-1})' & H_1(A^{n-2})' & \cdots & H_1 \end{bmatrix}\Pi\right\} + \Pi_o, \tag{29}$$

where

$$\Pi := \mathbb{E}\{\Xi'_{\boldsymbol{\theta}}\Delta\Xi_{\boldsymbol{\theta}} + \Xi'_{\boldsymbol{\theta}}\Delta K + K'\Delta\Xi_{\boldsymbol{\theta}}\}, \tag{30a}$$

$$\Pi_o := \mathrm{tr}\{\Sigma^o K'\Delta K\} + \mathrm{tr}\{\mathbb{E}\{\xi_{\boldsymbol{\theta}}\xi'_{\boldsymbol{\theta}}\}\Delta\} + G, \tag{30b}$$

$$\Sigma^o := \mathbb{E}\{\boldsymbol{x}^o(\boldsymbol{x}^o)'\} = \begin{bmatrix} \Sigma^o_n & A\Sigma^o_{n-1} & \cdots & A^{n-1}\Sigma^o_1 \\ \Sigma^o_{n-1}A' & \Sigma^o_{n-1} & \cdots & A^{n-2}\Sigma^o_1 \\ \vdots & \vdots & \ddots & \vdots \\ \Sigma^o_1(A^{n-1})' & \Sigma^o_1(A^{n-2})' & \cdots & \Sigma^o_1 \end{bmatrix},$$

and $H_k := \mathbb{E}\{\hat{\boldsymbol{x}}^o_k(\hat{\boldsymbol{x}}^o_k)'\}$, $\hat{\boldsymbol{x}}^o_k = \mathbb{E}\{\boldsymbol{x}^o_k|\boldsymbol{s}_{[1,k]}\}$ and $\boldsymbol{x}^o_k$ evolves according to (11). Hence, the optimization problem (29) faced by S can be written as an affine function of H_k's as follows[6]:

$$\min_{\substack{\eta_k\in\Upsilon,\\ k=1,\ldots,n}} \sum_{k=1}^{n}\mathrm{tr}\{V_kH_k\} + \Pi_o, \tag{31}$$

for certain symmetric deterministic matrices $V_k \in \mathbb{R}^{m\times m}$, $k = 1,\ldots,n$, which are given by

$$V_k := \Pi_{k,k} + \sum_{l=k+1}^{n}\Pi_{k,l}A^{l-k} + (A^{l-k})'\Pi_{l,k}, \tag{32}$$

and $\Pi_{k,l}$ is the corresponding $m\times m$ sub-block of Π. We note that the expectation in (30) is taken over all $O(t^N)$ scenarious. S can compute the expectation numerically through the Monte Carlo method [14].

Next, we aim to compute the solutions of the nonlinear (possibly non-convex) optimization problem (31) via an analytical approach instead of brute force approaches, e.g., particle swarm optimization [15], that would search n matrices with $m\times m$ dimensions over nm^2 dimensional space, i.e., $\mathbb{R}^{nm^2}$. To this end, similar to [5], we employ the approach in [16], which considers a semi-definite programming problem that bounds (31) from below, and then, computes strategies for the original problem, which can optimize the lower bound. Based on this, the following theorem characterizes equilibrium achieving secure sensor strategies.

Theorem 1. *The optimal linear secure sensor strategies can be computed via Algorithm 1, described in Table I.*

Proof. Note that (31) has identical compact form with equation (68) in [5] for different matrices Π and Π_o. Then, based on Lemma 3 in [16], the proof follows by the proof of Theorem 2 in [5]. Particularly, the lower bound is given by

$$\begin{aligned} &\min_{\substack{S_k\in\mathbb{S}^m,\\ k=1,\ldots,n}} \textstyle\sum_{k=1}^{n}\mathrm{tr}\{V_kS_k\} \quad \leq \quad \min_{\substack{\eta_k\in\Upsilon,\\ k=1,\ldots,n}} \textstyle\sum_{k=1}^{n}\mathrm{tr}\{V_kH_k\}, \\ &\text{s.t. } \Sigma^o_j \succeq S_j \succeq AS_{j-1}A'\ \forall j \end{aligned} \tag{33}$$

where $\Sigma^o_j = \mathbb{E}\{\boldsymbol{x}^o_j(\boldsymbol{x}^o_j)'\}$, $S_0 := O$. By Theorem 4 in [16], the solution of the lower bound in (33), $S^*_1,\ldots,S^*_n$, is given by

$$S^*_k = AS^*_{k-1}A' + (\Sigma^o_k - AS^*_{k-1}A')^{1/2}P_k(\Sigma^o_k - AS^*_{k-1}A')^{1/2}, \tag{34}$$

for $k = 1,\ldots,n$, where $S^*_0 = O$ and $P_k \in \mathbb{S}^m$ is a certain symmetric idempotent matrix. On the other hand, for given $\boldsymbol{s}_k = \mathscr{L}'_k\boldsymbol{x}_k$, H_k has the following recursion:

$$\begin{aligned} H_k = AH_{k-1}A' + (\Sigma^o_k - AH_{k-1}A')\mathscr{L}_k(\mathscr{L}'_k(\Sigma^o_k - AH_{k-1}A')\mathscr{L}_k)^{\dagger} \\ \times\mathscr{L}'_k(\Sigma^o_k - AH_{k-1}A'). \end{aligned} \tag{35}$$

Then, the optimal secure sensor strategies are given by

$$\mathscr{L}_k = (\Sigma^o_k - AS^*_{k-1}A')^{-1/2}U_k\Lambda_k. \tag{36}$$

where $U_k, \Lambda_k \in \mathbb{R}^{m\times m}$ are the matrices in the eigen decomposition of P_k in (34), i.e., $P_k = U_k\Lambda_kU'_k$. □

[5] Omitted steps are identical to the derivation of $\boldsymbol{u}^*_{\mathrm{F},k}$, which can be found in [5].

[6] H_k depends on the optimization arguments $\eta_{[1,n]}$ due to $\boldsymbol{s}_k = \eta_k(\boldsymbol{x}_k)$.

TABLE I: Detailed description of Secure Sensor Design Algorithm.

```
Algorithm 1: Secure Sensor Design
Compute $V_k$'s:
    Compute $K_{\mathrm{F},k}, \Delta_{\mathrm{F},k}$, and $K_{\mathrm{A}_i,k}$ for $k = 1, \ldots, n$ and $i = 1, \ldots, t$
      via (13) and (17).
    Compute $\Phi_{\mathrm{F}}^{(1)}$ by (12), $\Phi_{\mathrm{A}_i}^{(1)}$ by (16), and $F^{(1)}$ by (18).[7]
    For all $\theta_{[1,N]} \in \Omega$:
        Compute $T_{\theta_{[1,N]}}$ and $M_{\theta_{[1,N]}}$, given by (23).
        Compute $\Xi_{\theta_{[1,N]}}$ and $\xi_{\theta_{[1,N]}}$ via (27).
    Compute $\Pi$ and $\Pi_o$, given by (30), based on $\Xi_{\theta_{[1,N]}}, \xi_{\theta_{[1,N]}}$.
    Then, compute $V_k$, $k = 1, \ldots, n$, via (32).
SDP Problem:
    Solve the SDP problem on the left hand side of (33) through
      a numerical toolbox, e.g., CVX [17], [18], and obtain the
      solutions $S_k^*$, for $k = 1, \ldots, n$.
    Set $S_0^* = O$.
Optimal secure sensor strategies:
    Compute the corresponding idempotent matrices $P_k, \forall k$, by
      using $S_k^*$, $\forall k$, and (34).
    Compute the eigen decompositions: $P_k = U_k \Lambda_k U_k'$.
    Compute $\mathscr{L}_k$, $\forall k$, by using $S_{k-1}^*, U_k, \Lambda_k$, and (36).
    And $\eta_k(\boldsymbol{x}_k) = \mathscr{L}_k' \boldsymbol{x}_k$.
```

V. CONCLUSION

In this paper, we have introduced a secure sensor design framework for resiliency of cyber-physical systems prior to the attack detection. We have specifically considered LQG control systems, where the controller could have been compromised within the operation by various attackers with certain adversarial control objectives. The controller (and correspondingly the attacker when infiltrated into it) has access to the sensor outputs and the previous control inputs. Therefore, we have sought to design the linear sensor outputs cautiously by taking the possibility of undetected attacks into consideration. We have provided an algorithm to compute the secure sensor outputs that lead to the minimum damage in terms of system's quadratic control objective.

Some future directions of research on this topic include: identification of attacker objectives based on the previous control inputs so that the system can update (enhance) the belief about the underlying attack statistics based on such identifications and the scenarios where the sensor has partial or noisy observation of the state.

REFERENCES

[1] J. Giraldo, E. Sarkar, A. A. Cardenas, M. Maniatakos, and M. Kantarcioglu, "Security and privacy in cyber-physical systems: A survey of surveys," *IEEE Design & Test*, vol. 34, pp. 7–17, 2017.

[2] A. Humayed, J. Lin, F. Li, and B. Luo, "Cyber-physical systems security – A survey," *IEEE Internet of Things Journal*, vol. 4, no. 6, 2017.

[3] T. Başar and P. Bernhard, *H-infinity optimal control and related minimax design problems: A dynamic game approach*. Birkhäuser Basel, 1995.

[4] S. Han, M. Xie, H.-H. Chen, and Y. Ling, "Intrusion detection in cyber-physical systems: Techniques and challenges," *IEEE Systems Journal*, vol. 8, no. 4, pp. 1052–1062, 2014.

[5] M. O. Sayin and T. Başar, "Secure sensor design against undetected infiltrations: Minimum impact-minimum damage," *IEEE Trans. Automatic Control*, submitted for publication, available at ArXiv 1801.01630, 2018.

[6] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Trans. Information and System Security*, vol. 14, no. 1, 2009.

[7] Y. Chen, S. Kar, and J. M. F. Moura, "Cyber physical attacks with control objectives and detection constraints," in *Proc. 55th IEEE Conf. on Decision and Control (CDC)*, 2016, pp. 1125–1130.

[8] ——, "Cyber physical attacks constrained by control objectives," in *Proc. Americal Control Conference (ACC)*, 2016, pp. 1185–1190.

[9] R. Zhang and P. Venkitasubramaniam, "Stealthy control signal attacks in linear quadratic Gaussian control systems: Detectability reward tradeoff," *IEEE Trans. Inf. Forensics and Security*, vol. 12, no. 7, pp. 1555–1570, 2017.

[10] T. Cover and J. Thomas, *Elements of Information Theory*. John Wiley & Sons, 2012.

[11] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," *IEEE Trans. Autom. Control*, 2017.

[12] P. R. Kumar and P. Varaiya, *Stochastic systems: Estimation, identification and adaptive control*. Prentice Hall, Englewood Cliffs, NJ, 1986.

[13] M. O. Sayin and T. Başar, "Secure sensor design for cyber-physical systems against advanced persistent threats," in *Proc. Int. Conf. on Decision and Game Theory for Security on Lecture Notes in Computer Science*, S. Rass, B. An, C. Kiekintveld, F. Fang, and S. Schauder, Eds., vol. 10575. Vienna, Austria: Springer, Oct. 2017, pp. 91–111.

[14] D. P. Kroese, T. Brereton, T. Taimre, and Z. I. Botev, "Why the Monte Carlo method is so important today," *WIREs Comput Stat*, vol. 6, no. 6, pp. 386–392, 2014.

[15] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proc. IEEE Int. Conf. Neural Networks*, 1995, pp. 1942–1948.

[16] M. O. Sayin, E. Akyol, and T. Başar, "Hierarchical multi-stage Gaussian signaling games: Strategic communication and control," *Automatica*, submitted for publication, available at ArXiv 1609.09448, 2017.

[17] M. Grant and S. Boyd, "CVX: Matlab software for disciplined convex programming, version 2.1," http://cvxr.com/cvx, Mar. 2014.

[18] ——, "Graph implementations for nonsmooth convex programs," in *Recent Advances in Learning and Control*. Springer-Verlag Limited, 2008, pp. 95–110.

[7] For each type $\theta \in \Theta$, the matrices $\Phi_\theta^{(\kappa)}$ and $\underline{\Phi}_\theta^{(\kappa)}$ are sub-block matrices in $\Phi_\theta^{(1)}$ and can be computed via $\Phi_\theta^{(1)}$.