Automatica 93 (2018) 55–63

Contents lists available at ScienceDirect

Automatica

journal homepage: www.elsevier.com/locate/automatica

A hybrid stochastic game for secure control of cyber-physical systems☆

Fei Miao [a,*], Quanyan Zhu [c], Miroslav Pajic [d], George J. Pappas [b]

[a] University of Connecticut, Storrs, CT, USA
[b] University of Pennsylvania, Philadelphia, PA, USA
[c] New York University, Brooklyn, NY, USA
[d] Duke University, Durham, NC, USA

ARTICLE INFO

Article history:
Received 29 August 2016
Received in revised form 31 August 2017
Accepted 26 December 2017

Keywords:
Stochastic game
Secure control
Saddle-point equilibrium

ABSTRACT

In this paper, we establish a zero-sum, hybrid state stochastic game model for designing defense policies for cyber-physical systems against different types of attacks. With the increasingly integrated properties of cyber-physical systems (CPS) today, security is a challenge for critical infrastructures. Though resilient control and detecting techniques for a specific model of attack have been proposed, to analyze and design detection and defense mechanisms against multiple types of attacks for CPSs requires new system frameworks. Besides security, other requirements such as optimal control cost also need to be considered. The hybrid game model we propose contains physical states that are described by the system dynamics, and a cyber state that represents the detection mode of the system composed by a set of subsystems. A strategy means selecting a subsystem by combining one controller, one estimator and one detector among a finite set of candidate components at each state. Based on the game model, we propose a suboptimal value iteration algorithm for a finite horizon game, and prove that the algorithm results an upper bound for the value of the finite horizon game. A moving-horizon approach is also developed in order to provide a scalable and real-time computation of the switching strategies. Both algorithms aim at obtaining a saddle-point equilibrium policy for balancing the system's security overhead and control cost. The paper illustrates these concepts using numerical examples, and we compare the results with previously system designs that only equipped with one type of controller.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Cyber-Physical Systems (CPS) feature a tight integration of embedded computation, networks, controlled physical processes, and provide the foundation of critical infrastructures such as transportation systems, smart grids, water service systems and so on (Kim & Kumar, 2012). However, the integration structures also result in vulnerability under malicious attacks (Cardenas, Amin, Sionpoli, Perrig, & Sastry, 2009). Recoded incidents caused by attacks show that CPS attacks can disrupt critical infrastructures and lead to undesirable, catastrophic consequences (Slay & Miller, 2007). While cyber security tools have focused on prevention mechanisms, there are still challenges on how to leverage the ability of control systems to keep system resilient under a smart adversary.

Detection methods for various types of attacks have been analyzed in the literature. Pasqualetti, Dorfler, and Bullo (2013) propose a framework for attacks and monitors of CPS perspectives. Mo et al. (2012) analyze security challenges and countermeasures in smart grids. Pajic et al. (2014) present resilient state estimators for systems with noise and modeling errors. Humphreys (2013) analyzes spoofing attacks against cryptographically-secured Global Navigation Satellite System (GNSS) signals and detection strategies. Miao, Zhu, Pajic, and Pappas (2016) design a coding scheme for sensor outputs to detect stealthy data injection attacks over the communication channel.

☆ This material is based on research sponsored by DARPA under agreement number FA8750-12-2-0247. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA or the U.S. Government. This work was also supported in part by NSF CNS-1505701, CNS-1505799 grants, and the Intel-NSF Partnership for Cyber-Physical Systems Security and Privacy. The material in this paper was partially presented at the 52nd IEEE Conference on Decision and Control, December 10–13, 2013, Florence, Italy and the 53rd IEEE Conference on Decision and Control, December 15–17, 2014, Los Angeles, CA, USA. This paper was recommended for publication in revised form by Associate Editor Bert Tanner under the direction of Editor Christos G. Cassandras.

* Corresponding author.
E-mail addresses: fei.miao@uconn.edu (F. Miao), quanyan.zhu@nyu.edu (Q. Zhu), miroslav.pajic@duke.edu (M. Pajic), pappasg@seas.upenn.edu (G.J. Pappas).

https://doi.org/10.1016/j.automatica.2018.03.012
0005-1098/© 2018 Elsevier Ltd. All rights reserved.

In general, attack models are used as parameters to design defense schemes. However, a specific detection approach alone is not sufficient, when the system does not have knowledge which attack will happen among various types of potential attacks. CPS are usually resource constrained systems, which prevents running all available modules at the same time. Besides security, other requirements like optimal cost need to be addressed during control systems design. Consequently, considering control and defense costs with the effects of multiple attacks, strategic methods that balance the system performance and security requirements are necessary. In this work, we consider the case that at each time instant, only one detector is active because of the limits of resources. Our approach can be generalized to more than one detector being active at every time instance.

The application of game theory to security problems has raised a lot of interest in recent years. Selected works that apply game-theoretic approaches in computer networks security and privacy problems are summarized by Manshaei, Zhu, Alpcan, Basar, and Hubaux (2013). Zhu and Martinez (2011) propose a receding-horizon dynamic Stackelberg game model for systems under correlated jamming attacks. Zhu and Basar (2015) propose game-theoretic methods for robust and resilient control of CPSs. However, none of these works have considered switching policies under multiple types of attacks, with payoffs as functions of system dynamics and probabilistic detection rate.

Building a framework that captures the hybrid system dynamics and interactions with attacks is pivotal for security analysis and design of CPS. To achieve this goal, our first step is to establish a zero-sum hybrid stochastic game model. The hybrid state of the game model contains a dynamic system state that captures the evolution of the physical processes, and discrete cyber modes that represent different security states of the CPS according to information provided by the detector. Then a suboptimal value iteration algorithm is developed for the finite horizon hybrid stochastic game. Compared with our previous game model (Miao, Pajic, & Pappas, 2013) that only switches between two controllers against replay attacks and needs strategy history to calculate a strategy, in this work the hybrid state stochastic game strategy calculation process does not depend on the strategy history.

We then propose a moving-horizon computation methodology to reduce the computational complexity of finding a saddle-point equilibrium for the hybrid stochastic game. This is a scalable and computationally efficient algorithm. At each stage, the system selects a window of finite length for the physical state, and computes the stationary saddle-point strategies for the associated finite stochastic game, with the game state reformulated as the joint cyber and physical states. A preliminary result of the moving-horizon algorithm appeared in the conference paper Miao and Zhu (2014); in this journal version, we have included more details about different types of attacks and each element of the game model, revised analysis of the moving horizon algorithm compared with the suboptimal algorithm, and added more simulation results. The cost comparison with the suboptimal algorithm shows that the real-time algorithm does not sacrifice system performance much. The contributions of this work are summarized as follows:

(1) We formulate a zero-sum, hybrid stochastic game framework for designing a switching policy for a system under various types of attacks.
(2) We design a suboptimal algorithm for the finite horizon hybrid stochastic game, and prove that the algorithm provides an upper bound for the optimal cost of the system.
(3) We develop a real-time algorithm to reduce the computation overhead of the game model.

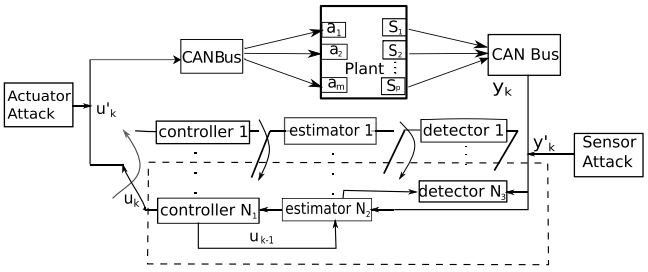


Fig. 1. Switching system diagram, where the system is equipped with N_1 controllers, N_2 estimators and N_3 detectors and switches among N subsystems. A subsystem (controller N_1, estimator N_2, and detector N_3) is chosen here.

This paper is organized as follows. We describe the system, attack models, and motivation of game-theoretic techniques for switching policies in Section 2. In Section 3, we formulate a zero-sum, hybrid stochastic game between the system and the attacker. A suboptimal algorithm for the finite horizon game is developed in Section 4. The moving horizon algorithm and its computational complexity are analyzed in Section 5. Section 6 compares the complexity and system performance of the finite horizon and the receding horizon algorithms. Finally, Section 7 provides concluding remarks.

2. Switched system and attack model

We consider the CPS security problem when both the system and attacker have limited knowledge about the opponent. The system is equipped with multiple controllers/estimators/detectors, such that each combination of these components constitutes a subsystem. A subsystem has a probability to detect specific types of attacks with different control and detection costs. To balance the security overhead and the control cost under various attacks, we consider switching among subsystems (choose a model for every component) according to the system dynamics and detector information. A switched system model is shown in Fig. 1, and the model of each component is described with a concrete example in the rest of this section. It is worth noting that the set of subsystems is not restricted and can be further generalized.

LTI plant and sensor attack model: Consider a class of LTI plants described by:

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{w}_k, \quad \mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, \tag{1}$$

where $\mathbf{x}_k \in \mathbb{R}^n$, $\mathbf{u}_k \in \mathbb{R}^p$ and $\mathbf{y}_k \in \mathbb{R}^m$ denote the discrete time state, input and output vectors respectively, and $\mathbf{w}_k \sim \mathcal{N}(0, \mathbf{Q})$, $\mathbf{v}_k \sim \mathcal{N}(0, \mathbf{R})$ are independent and identically distributed (IID) Gaussian random noise. The initial state is $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \Sigma)$. Sensors or the communication between sensors and estimators are vulnerable, and attacker can change values $\mathbf{y}_k$ that sent from sensors of system (1), and the compromised sensor measurements are defined as $\mathbf{y}'_k$ according to the types of attacks we consider. For instance, if the attacker can inject arbitrary data $\mathbf{y}^a_k$ to sensors, $\mathbf{y}'_k = \mathbf{y}_k + \mathbf{y}^a_k$; for replay attacks, the attacker can choose the replay window size T_2, let $\mathbf{y}'_k = \mathbf{y}_{k-T_2}$ and decide whether to send the delayed plant outputs at k.

Estimators: The physical dynamical state of the system is provided by an estimator, for instance, attack resilient estimator (Pajic et al., 2014), l_1 norm state estimator (Pajic, Tabuada, Lee, & Pappas, 2015), fault detection filter (Zhong, Ding, Lam, & Wang, 2003), or the widely applied Kalman filter. When $(\mathbf{A}, \mathbf{B})$ is stabilizable, $(\mathbf{A}, \mathbf{C})$ is detectable, a steady state Kalman filter exists.

Controllers: A state feedback control law is described as $\mathbf{u}_k = L(\hat{x}_{k|k})$, where $L(\cdot)$ is a linear function, $\hat{x}_{k|k}$ is the estimated state. Mo

and Sinopoli (2009) increase the detection rate by adding an IID Gaussian signal $\Delta\mathbf{u}_k \sim \mathcal{N}(0, \mathcal{L})$ to $\mathbf{u}_k^*$ to an optimal LQG controller as $\mathbf{u}_k = \mathbf{u}_k^* + \Delta\mathbf{u}_k$, and increase the control cost. Then always applying the non-optimal controller for detecting a replay attack is not cost optimal, especially when there is no replay at all during a long time.

Detectors: We assume that every detector of the subsystem provides a detection rate for a specific type of attack, and a system is equipped with several detectors in order to deal with multiple types of attacks. Researchers have designed probabilistic detectors with respect to different attacks. For instance, Zhong et al. (2003) design a fault detection filter, including a residual estimator and a threshold and a decision logic unit. Hypothesis testing strategies such as maximum likelihood (MLE), maximum a posteriori (MAP), and minimum mean square error (MMSE) account for GPS spoofing attack is presented by Humphreys (2013).

Cyber state—discrete modes of the system: We denote the modes of a vulnerable system as three constants $S = \{\delta_1, \delta_2, \delta_3\}$. State $\delta_1 = \textit{safe}$ describes that the system has already successfully detected an attack; $\delta_2 = \textit{no detection}$ specifies that the alarm is not triggered; finally, the system enters state $\delta_3 = \textit{false alarm trigger}$ when the alarm is triggered while no attack has yet occurred. The mode depends on the probability detection rate. We assume that once the alarm is triggered, the system will stop the execution and decide whether to react to occurred attacks or it is a false alarm.

3. A hybrid stochastic game model

To obtain a switching policy that minimizes the expected real-time worst case payoff for the given subsystems, we formulate a zero-sum, hybrid stochastic game between the system and the attacker. System dynamics knowledge are combined with the game definition, and the quantitative process for the game parameters will be introduced in this section. We assume that one game stage k is also one time step of the physical system. The total stage number is K. The hybrid game state space $(X_{[k-T,k]} \times S)$ contains information about both the system dynamics $\mathbf{x}_k$ and the discrete modes δ_l, $l = 1, 2, 3$. Here, T is the window size of system dynamics needed to keep the state transition between stages k and $(k+1)$ Markov. The joint state includes information we need to compute the game strategy at the current stage. This is the main difference compared with the previous work (Miao et al., 2013), while the latter is not Markov since it needs to consider all the possible histories of strategies for deciding the physical dynamics and getting a strategy. At each stage $k \in \{T, \ldots, K+T\}$, parameters include the action space for the attacker (system) A_t (A_s), the state transition probability matrix $\mathbb{P}_k$, and the immediate payoff matrix r_k. The solution set of the game is mixed strategies $\mathbf{F}_k$ for the attacker, and $\mathbf{G}_k$ for the system. Formally, the game is defined as a sequence of tuples: $\{(X_{[k-T,k]} \times S), A_t, A_s, \mathbf{F}_k, \mathbf{G}_k, P, r\}$.

Game State Space: The joint state of the system at stage k is described by the pair $s_{kl} = (x_{[k-T,k]}, \delta_l)$, where

$$x_{[k-T,k]} = (x_{k-T}, x_{k-T+1}, \ldots, x_k) \in X_{[k-T,k]}$$

is the discrete-time dynamics of the physical process provided to the system—the state estimations $\hat{x}_{k-T}, \ldots, \hat{x}_k, \delta_l \in S = \{\delta_1, \delta_2, \delta_3\}$ denote the cyber state of the system. We assume that once the game reaches δ_1, the system wins and will not enter other modes till next game, i.e., δ_1 is an absorbing state. The moving-horizon transition of the joint states on stage axis is shown as Fig. 2. The window size of system dynamics T keeps the state transition between time k and $k+1$ Markov. For instance, if the detector of the system requires system dynamics $\hat{x}_{[k-T_1,k]}$, and we consider sensor

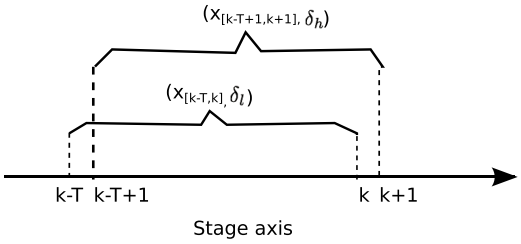


Fig. 2. Joint state transition of the hybrid stochastic game when moving the horizon of game state one step ahead. When the state transits from stage k to $k+1$, we slice the window of the sequence of physical dynamics one step ahead, add x_{k+1} and remove x_{k-T}, thus $x_{[k-T,k]} \rightarrow x_{[k-T+1,k+1]}$. The piecewise constant modes δ_l, δ_h describe the cyber states provided by the detector at stage k, respectively.

data injection attacks and replay attacks with replay windows less than T_2 steps, then $T = max\{T_1, T_2\}$.

Attacker's Action Space: We assume that the system is vulnerable to different attack models described by the action space A_t, where

$$A_t = \{a_1(x_{[k-T,k]}), a_2(x_{[k-T,k]}), \ldots, a_M(x_{[k-T,k]})\}$$

is the attacker's action space at stage k, and a_1 means no attack. Here we only consider discretized action space of the attacker for computational efficiency. For the LTI system dynamics considered in this work, the distance of a continuous point to its nearest discrete point in action space is bounded. With bounded error of the dynamics by discretized continuous action space, the quality of game solutions under different conditions is analyzed by work (Kroer & Sandholm, 2015).

The actions can describe both multiple types of attacks and the same type attack with different values. For instance, when considering only sensor data injection attacks with different norms of injection value, we will denote $a_i(x_{[k-T,k]}), i = 2, 3, \ldots$ as changing the sensor value from $\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$ to $\mathbf{y}'_k = \mathbf{y}_k + \mathbf{y}^a_{k,i}$, where any injection $\mathbf{y}^a_k$ is classified as $a_i(x_{[k-T,k]}), i = inf\{i : \|\mathbf{y}^a_k - \mathbf{y}^a_{k,i}\|_2\}$ in attacker's action space. Similarly, for replay attack only, the action space is discretized as changing sensor values from $\mathbf{y}_k = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k$ to $\mathbf{y}'_k = \mathbf{y}_{k-T_i}$ for action index $a_i(x_{[k-T,k]})$, where any replay time length T_a is classified as $a_i(x_{[k-T,k]}), i = inf\{i : |T_a - T_i|\}$. Considering multiple types of attacks, we assume that the system is vulnerable under m_a types of attacks, and attack type A_i is corresponding to $M_{a,i}$ discretized actions in the action space, then there are $\sum_{i=1}^{m_a} M_{a,i} + 1$ actions in total within the attacker's action space A_t.

System's Action Space: The system's action space at stage k is defined as

$$A_s = \{u_1(x_{[k-T,k]}), u_2(x_{[k-T,k]}), \ldots, u_N(x_{[k-T,k]})\},$$

where u_j is the index for the jth subsystem. We assume that the N subsystems (a model for each component in Fig. 1) are determined priorly. For example, a subsystem can be the plant with a given optimal LQG controller, a Kalman filter and a χ^2 detector. A subsystem can also be the plant with an optimal LQG controller, a resilient state estimator (Pajic et al., 2014) and its corresponding estimation residual checking component. We assume that the attacker's action space is defined, with corresponding system's action or a subsystem that the detection rate is greater than 0. A switched system does not ensure performance under the attack outside the action space of the game.

Mixed Strategy: Let $f^i_k(s_{kl})$ ($g^j_k(s_{kl})$) be the probability that the attacker (system) chooses action $a_i(x_{[k-T,k]}) \in A_t$ ($u_j(x_{[k-T,k]}) \in A_s$) at state $s_{kl} \in (X_{[k-T,k]} \times S)$. Define $\mathbf{F}_k$ and $\mathbf{G}_k$ as the mixed strategy sets of the attacker and the system for stage k: $\mathbf{F}_k :=$

$\{\mathbf{f}_k = [\mathbf{f}_k(s_{k1}), \mathbf{f}_k(s_{k2}), \mathbf{f}_k(s_{k3})] | f_k^i(s_{kl}) \geq 0, \mathbf{f}_k \in [0,1]^{M\times 3}, \sum_{a_{ik}\in A_{tk}} f_k^i(s_{kl}) = 1, \mathbf{f}_k(s_{kl}) \in \mathbb{R}^M, \forall s_{kl} \in (X_{[k-T,k]} \times S)\}$, $\mathbf{G}_k := \{\mathbf{g}_k = [\mathbf{g}_k(s_{k1}), \mathbf{g}_k(s_{k2}), \mathbf{g}_k(s_{k3})] | g_k^j(s_{kl}) \geq 0, \mathbf{g}_k \in [0,1]^{N\times 3}, \sum_{u_{jk}\in A_{sk}} g_k^j(s_{kl}) = 1, \mathbf{g}_k(s_{kl}) \in \mathbb{R}^N, \forall s_{kl} \in (X_{[k-T,k]} \times S)\}$.Note that $\mathbf{x}_{[k-T,k]}$ provides exogenous information for the strategy $\mathbf{f}_k(\mathbf{g}_k)$, since for every l, $\mathbf{f}_k(s_{kl})(\mathbf{g}_k(s_{kl}))$ is the strategy at mode δ_l for the same $\mathbf{x}_{[k-T,k]}$ at stage k. Hence, $\mathbf{g}_k$ and $\mathbf{f}_k$ are finite dimensional vectors, that the stationary strategy chosen by each player at stage k depends on the cyber state.

System and Subsystem Dynamics under game framework: Given the subsystem and attack models in Section 2 and the game definition, we show the dynamics at stage k given an action pair $(a_i(x_{[k-T,k]}), u_j(x_{[k-T,k]}))$ (assume initial $\hat{\mathbf{x}}_{1|0} = \bar{\mathbf{x}}_0$, $\mathbf{x}_1 = \mathbf{x}_0$). Each action pair $(a_i(x_{[k-T,k]}), u_j(x_{[k-T,k]}))$ defines the corresponding system dynamics at k. For instance, when we focus on sensor attacks (like replay or false data injection), let $\boldsymbol{\gamma}_k(a_i(x_{[k-T,k]}), u_j(x_{[k-T,k]}))$ be the control input with $(a_i(x_{[k-T,k]}), u_j(x_{[k-T,k]}))$, a subsystem $u_j(x_{[k-T,k]})$ with a Kalman filter, an optimal LQG controller has the following dynamics (we denote $(a_i(x_{[k-T,k]}), u_j(x_{[k-T,k]}))$ as (a_{ik}, u_{jk}) for convenience):

$$
\begin{aligned}
&\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1},\\
&\mathbf{y}_k = \begin{cases} a_{1k} = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, & \text{without attack}\\ a_{ik},\ i = 2,\ldots,M, & \text{with attack,}\end{cases}\\
&\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1},\\
&\hat{\mathbf{x}}_{k|k}(a_{ik}) = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(a_{ik} - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}),\\
&\hat{\mathbf{x}}_{k+1|k}(a_{ik}, u_{jk}) = \mathbf{A}\hat{\mathbf{x}}_{k|k}(a_{ik}) + \mathbf{B}\boldsymbol{\gamma}_k(a_{ik}, u_{jk}),\\
&\boldsymbol{\gamma}_k(a_{ik}, u_{jk}) = \mathbf{L}\hat{\mathbf{x}}_{k|k}(a_{ik}),\\
&\mathbf{z}_{k+1}(a_{ik}, u_{jk}) = a_{ik} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}(a_{ik}, u_{jk}).
\end{aligned}
\tag{2}
$$

State Transition Probability: Given a set of subsystem models, define the state transition probability P as a function of the state of the game and both players' actions $P : (X_{[k-T,k]} \times S) \times A_t \times A_s \to [0,1]$, where

$$P(s_{(k+1)h}|s_{kl}, a_{ik}, u_{jk}), \quad h = 1,2,3$$

is the probability that system transits from state s_{kl} to state $s_{(k+1)h}$ at stage $k+1$, given both players' action (a_{ik}, u_{jk}) at stage k. Given the current game state $s_{kl} = (x_{[k-T,k]}, \delta_l)$ and an action pair (a_{ik}, u_{jk}), the dynamics of the system at stage $k+1$ is described as $x_{[k-T+1,k+1]}$ for all possible cyber modes $\delta_h \in S$, hence the dimension of state transition probability $P(s_{(k+1)h}|s_{kl}, a_{ik}, u_{jk})$ is determined by the number of cyber modes of the game. We denote $P(s_{(k+1)h}|s_{kl}, a_{ik}, u_{jk})$ as $P^{ij}(s_{(k+1)h}|s_{kl})$ for short. As a state transition probability, this function should also satisfy

$$
\begin{aligned}
&\sum_{\delta_h\in S} P^{ij}(s_{(k+1)h}|s_{kl}) = 1, \quad \forall (a_{ik}, u_{jk}) \in A_t \times A_s,\\
&s_{(k+1)h} \in (X_{[k-T+1,k+1]} \times S),\ s_{kl} \in (X_{[k-T,k]} \times S).
\end{aligned}
$$

The transition probability is provided by intrusion detectors of the subsystem.

Immediate Payoff Function: The immediate payoff matrix at stage k is a $\mathbb{R}^{M\times N}$ matrix for given game state and every action pair (a_{ik}, u_{jk}). We define the immediate payoff function as a continuous, convex function of the hybrid game state and the actions of both players

$$r : (X_{[k-T,k]} \times S) \times A_t \times A_s \to \mathbb{R}^{M\times N},$$

where $r(s_{kl}, a_{ik}, u_{jk}) \geqslant 0$ is the payoff at joint state s_{kl} given action pair (a_{ik}, u_{jk}). For definition convenience, we denote $r(s_{kl}, a_{ik}, u_{jk})$ as $r^{ij}(s_{kl})$ for short, since it is the element on the ith row and jth column of the payoff matrix $r(s_{kl})$. It is a zero-sum game between the system and the attacker, and we assume the system is the minimizer and the attacker is the maximizer, hence the payoff function for the attacker and the system is defined as

$$r^{ij}(s_{kl}) = r_t^{ij}(s_{kl}) = -r_s^{ij}(s_{kl}).$$

For instance, when the linear quadratic cost is a metric of system performance, let $\gamma_k(a_{ik}, u_{jk})$ be the control input given action pair (a_{ik}, u_{jk}), then the payoff function is defined as

$$
\begin{aligned}
r^{ij}(s_{k1}) &= \mathbb{E}[\hat{\mathbf{x}}_k^T]\mathbf{W}\mathbb{E}[\hat{\mathbf{x}}_k] + \mathbb{E}[\boldsymbol{\gamma}_k^T(a_{1k}, u_{jk})]\mathbf{U}\mathbb{E}[\boldsymbol{\gamma}_k(a_{1k}, u_{jk})],\\
r^{ij}(s_{k2}) &= \mathbb{E}[\hat{\mathbf{x}}_k^T]\mathbf{W}\mathbb{E}[\hat{\mathbf{x}}_k] + \mathbb{E}[\boldsymbol{\gamma}_k^T(a_{ik}, u_{jk})]\mathbf{U}\mathbb{E}[\boldsymbol{\gamma}_k(a_{ik}, u_{jk})],\\
r^{ij}(s_{k3}) &= p_f,
\end{aligned}
\tag{3}
$$

where p_f is the false alarm trigger penalty, the cost that the system needs to stop execution, check the reason of an alarm, and restart later; $\mathbf{x}_k$ is the physical state under the game framework. At mode δ_1 the system wins, so the payoff is a normal system payoff with correct sensor data. The larger p_f is, the less probable it is for the system to choose a strategy to transit to state s_{k3}.

System dynamics update with strategies at stage k: Let $p(s_{kl})$ be the probability that system is at state s_{kl} at stage k. The initial state distribution $p(s_{1l})$ is given. With a strategy $\mathbf{f}_k$, $\mathbf{g}_k$, the attacker and the system randomly sample an action pair (a_{ik}, u_{jk}) according to the probability distribution. Then, the control input and sensor value for calculating expectation cost are:

$$
\begin{aligned}
\mathbf{u}_k &= \sum_{j=1}^{N}\sum_{i=1}^{M}\sum_{l=1}^{3} p(s_{kl}) f_k^i(s_{kl}) g_k^j(s_{kl}) \boldsymbol{\gamma}_k(a_{ik}, u_{jk}),\\
\mathbf{y}_k &= \sum_{i=1}^{M}\sum_{l=1}^{3} p(s_{kl}) f_k^i(s_{kl}) a_{ik}.
\end{aligned}
$$

The probability that system is at state $s_{(k+1)h}$ for $k+1$ is:

$$p(s_{(k+1)h}) = \sum_{l=1}^{3} p(s_{kl})[\mathbf{f}_k(s_{kl})]^T P_k(s_{(k+1)h}|s_{kl})\mathbf{g}_k(s_{kl}).$$

4. Existence of an optimal strategy and suboptimal algorithm for a finite game

Based on the game formulation, in this section we discuss the existence of an optimal solution for the finite form of the hybrid stochastic game, and present an algorithm to compute a suboptimal system strategy.

4.1. Existence of the system's optimal strategy

We define the concatenation of strategies for K-stage game of each player ($\mathbf{f}$ for attacker and $\mathbf{g}$ for system) as $\mathbf{f} = \mathbf{f}_1 \cdots \mathbf{f}_K$, $\mathbf{f}_k \in \mathbf{F}_k$, $\mathbf{f} \in \mathbf{F}$, $\mathbf{g} = \mathbf{g}_1 \cdots \mathbf{g}_K$, $\mathbf{g}_k \in \mathbf{G}_k$, $\mathbf{g} \in \mathbf{G}$, $k = 1,2,\ldots,K$.

Definition 1. Let the random variable ζ_k describe the discrete state of the hybrid game at stage k, we define the conditional expected total payoff till $\tilde{K}$ for any $\mathbf{f}$, $\mathbf{g}$ as

$$
\begin{aligned}
&R_{\tilde{K}}(s, \mathbf{f}, \mathbf{g})\\
&= \sum_{k=1}^{\tilde{K}}\sum_{l=1}^{3} p(\zeta_k = \delta_l | \zeta_1 = s)[\mathbf{f}_k(s_{kl})]^T \tilde{r}_k(s_{kl})\mathbf{g}_k(s_{kl}),
\end{aligned}
$$

where $p(\zeta_k = \delta_l|\zeta_1 = s)$ is the probability that the discrete state of the hybrid game is δ_l at stage k given its initial discrete state $\zeta_1 = s$.

Since the immediate payoff of each stage satisfies that $0 \leq \tilde{r}_k^{ij}(s_{kl}) < \infty$, for all k, i, j, we have that $R_{\tilde{K}}(s, \mathbf{f}, \mathbf{g})$ is a nonnegative real-valued, nondecreasing function with $\tilde{K}$. Furthermore, for finite K

$$R_K(s, \mathbf{f}, \mathbf{g}) < \infty, \forall s \in S, \mathbf{f} \in \mathbf{F}, \mathbf{g} \in \mathbf{G}. \tag{4}$$

Similarly as the definition of value and optimal strategy for a zero-sum, finite discrete state, finite stage stochastic game, we define the value and optimal strategy for the hybrid state stochastic game defined in this work as the following.

Definition 2. A two-person zero-sum K-stage stochastic game is said to have a value vector v_K^* if $v_{K,s}^* = \underline{v}_{K,s} = \bar{v}_{K,s}$, for any initial cyber state $s \in S$, where

$$\underline{v}_{K,s} = \sup_{\mathbf{f}\in\mathbf{F}} \inf_{\mathbf{g}\in\mathbf{G}} R_K(s, \mathbf{f}, \mathbf{g}),$$

$$\bar{v}_{K,s} = \inf_{\mathbf{g}\in\mathbf{G}} \sup_{\mathbf{f}\in\mathbf{F}} R_K(s, \mathbf{f}, \mathbf{g}).$$

For the finite value K-stage stochastic game, strategies $\mathbf{g}^*$ and $\mathbf{f}^*$ are called optimal at the saddle-point equilibrium for player two (the system) and player one (the attacker), respectively, if for all $s \in S$,

$$v_{K,s}^* = \inf_{\mathbf{g}\in\mathbf{G}} R_K(s, \mathbf{f}^*, \mathbf{g}), \quad v_{K,s}^* = \sup_{\mathbf{f}\in\mathbf{F}} R_K(s, \mathbf{f}, \mathbf{g}^*).$$

The game defined in this paper has finite action spaces, finite strategy space, finite discrete cyber modes and satisfies (4) with bounded total payoff in finite horizon. Therefore, there exists the value of the considered game and a saddle-point equilibrium or optimal strategy for the system shown in Basar and Olsder (1998).

4.2. Suboptimal algorithm for the finite game

Existing value iterative algorithms or dynamic programming algorithms for finite stochastic games cannot be used to solve the finite hybrid stochastic game defined in this work, since the discrete time dynamics $x_{[k-T,k]}$ of the game at stage k depends on that of the stage $k-1$, which is only available in the future algorithm iterations. Hence, we design a suboptimal algorithm based on the value iteration method for a finite horizon, finite discrete state stochastic game (Kearns, Mansour, & Singh, 2000) and robust game techniques (Aghassi & Bertsimas, 2006). The value iteration algorithm for a finite horizon, discrete state stochastic game (with fixed payoff r and state transition probability P at every stage) works in the way that if a player knew how to play in the game optimally from the next stage on, then, at the current stage, he would play with such strategies. The value of K-stage game is finally provided by the last step of iteration.

For a multi-stage game, to calculate the game value, we define the auxiliary matrix at stage k for every cyber state δ_l with system dynamics $x_{[k-T,k]}$ as $Q(s_{kl}) \in \mathbb{Q}_k \subset \mathbb{R}^{M\times N}$, and each element of $Q(s_{kl})$ for action pair (a_{ik}, u_{jk}) is defined as

$$\begin{aligned} &Q^{ij}(s_{kl}) \\ &= r^{ij}(s_{kl}) + \sum_{\delta_h\in S} P^{ij}(s_{(k+1)h}|s_{kl}) \cdot v_{k+1}(s_{(k+1)h}), \end{aligned} \tag{5}$$

where $v_{k+1}(s_{(k+1)h})$ is the game value from stage $k+1$, state $s_{(k+1)h}$ (with cyber mode δ_h) to the final stage K. For the final stage K, we define $Q(s_{Kl}) = r(s_{Kl})$. We define a one-shot game at stage k as a finite action space, zero-sum game between the system and the attacker with payoff matrix $Q(s_{kl})$, i.e., $Q^{ij}(s_{kl})$ is the payoff for action pair (a_{ik}, u_{jk}) of stage k. In each one-shot game, the system only considers a strategy $f_k(s_{kl})$ to minimize the worst case payoff caused by the attacker according to matrix $Q(s_{kl})$. Here $Q(s_{kl})$ is defined based on the system dynamics and the state transition probability provided by the detector. An alternative algorithm with unknown transition matrix or payoffs will be our future work.

Similarly as the value iteration algorithm for a discrete state stochastic game (Kearns et al., 2000), Algorithm 1 of the finite hybrid state stochastic game starts from the last stage, then gets the optimal one-stage strategy and the upper bound of game value at each stage. By calculating values of all stages until backwards to the first stage, Algorithm 1 returns an upper bound for the value of the total payoff in K-stages.

To estimate the values at each step, we consider the immediate payoff $r(s_{kl})$, the state transition probability $P(s_{(k+1)h}|s_{kl})$ and the game value estimated at the previous step uncertain parameters for the one shot robust game (Aghassi & Bertsimas, 2006). Then approximate each iteration value as the value of the robust one shot zero sum game. Algorithm 1 provides an upper bound for the game value and the corresponding suboptimal strategy for the system. The idea is to solve a robust game at each iteration step—i.e., minimize the worst-case caused by extreme points of the set of auxiliary matrix $\mathbb{Q}_k$ defined for all possible dynamics $x_{[k-T,k]}$.

To quantify the boundary of the set of auxiliary matrix $\mathbb{Q}_k$ we need the expected values of system dynamics $\mathbf{x}_k, \mathbf{u}_k, \mathbf{y}_k, k = 1, \ldots, K$ defined in Eqs. (2), which is determined by the strategies from stage 1 till stage k. We first analyze the uncertain sets of the immediate payoff function at stage k, and the extreme points for the uncertain set $\mathbb{Q}_k$ depend on pure strategies . Let $\mathbf{f}_{k-1}^p, \mathbf{g}_{k-1}^p$ be the concatenation of previous pure strategies of the attacker and the system till stage $k \geqslant 2$, respectively, where

$$\mathbf{f}_{k-1}^p = \mathbf{f}_1^p \cdots \mathbf{f}_{k-1}^p, \quad \mathbf{g}_{k-1}^p = \mathbf{g}_1^p \cdots \mathbf{g}_{k-1}^p$$

satisfies that all $\mathbf{f}_t^p(s)$ $(\mathbf{g}_t^p(s))$ for $t = 1, 2, \ldots, k$ have only one non-zero element, i.e., the player chooses the corresponding action or the *pure* strategy.

Define a pure strategy auxiliary matrix $Q^p(s_{kl}) \in \mathbb{Q}_k^p$ as:

$$\begin{aligned} &Q^p(s_{kl}) \\ &= r^p(s_{kl}) + \sum_{\delta_h\in S} P^p(s_{(k+1)h}|s_{kl}) \cdot \bar{v}_{k+1}^p(s_{(k+1)h}), \end{aligned} \tag{6}$$

for stages $k = 1, \ldots, K-1$, and for the final stage $k = K$,

$$Q^p(s_{Kl}) = r^p(s_{Kl}). \tag{7}$$

For each stage k, $\bar{v}_k^p(s_{kl})$ is defined as

$$\bar{v}_k^p(s_{kl}) = \max_{Q^p(s_{kl})\in\mathbb{Q}_k^p} v^*[Q^p(s_{kl})], \tag{8}$$

where v^* is the function that yields the value of a zero-sum matrix game. Then the value $\bar{v}_{k+1}^p(s_{(k+1)h}) \geq 0$ to calculate the auxiliary matrix (6) is the upper bound of robust game value from stage $k+1$ till stage K, resulting from the iteration at stage $k+1$. This value iteration process is the key idea of the following Algorithm 1.

Algorithm 1 (*Suboptimal Algorithm for A Finite Hybrid Stochastic Game*)**.**

Input: System model parameters and game parameters.

Initialization: Compute the set of $\mathbb{Q}_k^p$ for every stage $k = T, \ldots, T+K$ given $\hat{\mathbf{x}}_{[0,T]}$; get the robust game value and corresponding strategies at stage K: $Q^p(s_{(K+T)l}) = r^p(s_{(K+T)l})$, $f^*(s_{(K+T)l})$, $g^*(s_{(K+T)l})$, $\bar{v}_{K+T}^p(s_{(K+T)l}) \leftarrow \pi(Q^p(s_{(K+T)l}))$.

Iteration: For $k = (K+T-1), \ldots, T$, obtain a set of auxiliary matrices $\mathbb{Q}_k^p$ for all $\mathbf{f}_k^p, \mathbf{g}_k^p$, where each matrix is defined in (6), then calculate:

$f^*(s_{kl}), g^*(s_{kl}), \bar{v}_k^p(s_{kl}) \leftarrow \pi(Q^p(s_{kl}))$

$\mathbf{f}_k^* = [\mathbf{f}^*(s_{kl}), l = 1, 2, 3], \quad \mathbf{g}_k^* = [\mathbf{g}^*(s_{kl}), l = 1, 2, 3].$

Return:strategies $\mathbf{f}_a = \mathbf{f}_T^* \cdots \mathbf{f}_{K+T}^*$, $\mathbf{g}_a = \mathbf{g}_T^* \cdots \mathbf{g}_{K+T}^*$ and the value upper bound $\bar{v}_1^p(s_{1l}), l = 1, 2, 3$.

Now consider the iteration for calculating $\bar{v}_k^p(s_{kl})$ from all matrix games $Q^p(s_{kl}) \in \mathbb{Q}_k^p$ applying Algorithm 1. We define any strategy concatenations to stage $k-1$ with at most one non-pure strategy at stage $(k-1)$ as

$$\begin{aligned} \mathbf{f}_{k-1}^{np} &= \mathbf{f}_{k-2}^{p}\mathbf{f}_{k-1}, \quad \mathbf{f}_{k-1} \in \mathbf{F}_{k-1}, \\ \mathbf{g}_{k-1}^{np} &= \mathbf{g}_{k-2}^{p}\mathbf{g}_{k-1}, \quad \mathbf{g}_{k-1} \in \mathbf{G}_{k-1}, \end{aligned} \tag{9}$$

where $\mathbf{f}_{k-2}^p, \mathbf{g}_{k-2}^p$ are concatenations of pure strategies to stage $(k-1)$. We denote the corresponding auxiliary matrix as $\tilde{Q}(s_{kl}) \in \tilde{\mathbb{Q}}_k$ for cyber state δ_l, the one shot game value based on payoff matrix $\tilde{Q}(s_{kl})$ as $\tilde{v}_k(s_{kl})$, i.e.,

$$\begin{aligned} &\tilde{Q}(s_{kl}) \\ &= \tilde{r}(s_{kl}) + \sum_{\delta_h \in S} \tilde{P}(s_{(k+1)h}|s_{kl}) \cdot \tilde{v}_{k+1}(s_{(k+1)h}). \end{aligned} \tag{10}$$

Here each possible hybrid state s_{kl} for time instant k is calculated from a none pure strategy defined as (9). Similarly, the value is defined as

$$\tilde{v}_k(s_{kl}) = \max_{\tilde{Q}(s_{kl}) \in \tilde{\mathbb{Q}}_k} v^*[\tilde{Q}(s_{kl})]. \tag{11}$$

The following theorem shows that at every stage k, $\bar{v}_k^p(s_{kl})$ is greater than or equal to $\tilde{v}_k(s_{kl})$.

Theorem 3. *Consider the value iteration for stage k as a one shot robust game. Based on $\bar{v}_k^p(s_{kl}) \geq 0$ of previous iteration, we define the robust game value obtained at k as (8). Then for $k = 2, \ldots, K$, $\tilde{v}_k(s_{kl})$ (11) is upper bounded by $\bar{v}_k^p(s_{kl})$, i.e., $\tilde{v}_k(s_{kl}) \leqslant \bar{v}_k^p(s_{kl})$.*

Proof. Since $\bar{v}_{k+1}^p(s_{(k+1)h})$ is a nonnegative scalar value, the extreme points of the set $\mathbb{Q}_k$ are a subset of the extreme points of set $\mathbb{Q}_k^p$. Hence, by considering the value of matrix game $Q^p(s_{kl}) \in \mathbb{Q}_k^p$ defined in (6), we will get the upper bound of the maximum game value from extreme points of $\mathbb{Q}_k$.

Consider the following optimization problem for the system with constraint inequality (13) for any possible attacker's strategy vector $\mathbf{f}$ at each stage k

$$\min_{\mathbf{g}} \quad z \tag{12}$$

$$\text{subject to} \quad z \geq \max_{\tilde{Q}(s_{kl}) \in \tilde{\mathbb{Q}}_k} \mathbf{f}^T[\tilde{Q}(s_{kl})]\mathbf{g}. \tag{13}$$

As proven by Lemma 5 in Aghassi and Bertsimas (2006), (13) is equivalent to the following constraint that considers only the extreme points

$$z \geq \max_{Q^p(s_{kl}) \in \mathbb{Q}_k^p} \mathbf{f}^T[Q^p(s_{kl})]\mathbf{g}. \tag{14}$$

For the worst-case f, the above is also true. Hence, let

$$v_k^p(s_{kl}) = \max_{Q^p(s_{kl}) \in \mathbb{Q}_k^p} \min_{\mathbf{g}} \max_{\mathbf{f}} \mathbf{f}^T[Q^p(s_{kl})]\mathbf{g}. \tag{15}$$

For optimal policies $\mathbf{f}^*(s_{kl})$ and $\mathbf{g}^*(s_{kl})$, the above optimization problem (15) results in a cost

$$\max_{Q^p(s_{kl}) \in \mathbb{Q}_k^p} v^*[Q^p(s_{kl})].$$

However, $(\mathbf{f}^*(s_{kl}), \mathbf{g}^*(s_{kl}))$ can be non-pure strategies, meaning that when we apply $(\mathbf{f}^*(s_{kl}), \mathbf{g}^*(s_{kl}))$ to calculate system dynamics such as Eqs. (2), they will not result in any extreme point of set $\mathbb{Q}_{k+1}$.

Now consider the final stage K, we have

$$Q^p(s_{Kl}) = r^p(s_{Kl}), \ \tilde{Q}(s_{Kl}) = \tilde{r}(s_{Kl}),$$

and use the $Q^p(s_{Kl})$ and $\tilde{Q}(s_{Kl})$ in the above proof, value $\tilde{v}_k(s_{Kl})$ from $\tilde{Q}(s_{Kl})$ is smaller than $\bar{v}_k^p(s_{Kl})$ from the extreme points auxiliary matrix $Q^p(s_{Kl})$, i.e., for K, the following inequality holds

$$\tilde{v}_k(s_{Kl}) \leqslant \bar{v}_k^p(s_{Kl}).$$

Then, by induction, with the value $\tilde{v}_{k+1}(s_{(k+1)h})$ of iteration for stage $k+1$, $2 \leqslant k \leqslant K-1$ satisfies

$$\tilde{v}_{k+1}(s_{(k+1)h}) \leqslant \bar{v}_{k+1}^p(s_{(k+1)h}),$$

and nonnegative payoff and state transition probability $r_k^{ij} \geqslant 0$ and $\tilde{P}_k^{ij} \geqslant 0$, replacing $\tilde{v}_{k+1}(s_{(k+1)h})$ by $v_{k+1}^p(s_{(k+1)h})$ in (6) will make every entry of matrix $\tilde{Q}(s_{kl})$ smaller than matrix $Q^p(s_{kl})$. With a similar argument in the next iteration for stage $k-1$, we have

$$\tilde{v}_k(s_{kl}) \leqslant \bar{v}_k^p(s_{kl}).$$

Based on the above observation, we arrive at the suboptimal algorithm to compute the equilibrium solutions, illustrated in Algorithm 1. Note that for keeping the physical state $x_{[k-T,k]}$ of the first stage of the game starts at $\hat{x}_0$, in Algorithm 1 the K-stage game starts at $k = T$. This does not affect our proofs in this section for considering $k = 1, \ldots, T$. According to Theorem 3, we use Algorithm 1 to compute an upper bound of the value and the corresponding suboptimal strategy for every step. The function π computes the strategy and robust value as defined in (8).

The values of the finite stage game $\tilde{v}_k(s_{kl})$ and $\bar{v}_k^p(s_{kl})$ resulting from two auxiliary matrices $\tilde{Q}(s_{kl})$ $Q^p(s_{kl})$ are based on strategy concatenations that only differ at stage $k-1$ (i.e., the same and pure strategies from stages 1 to $(k-2)$). By value iteration backward to stage 1, we compare the game value for all possible strategies and the robust game value $\bar{v}_1^p(s_{1l})$ of Algorithm 1 in the following theorem.

Corollary 4. *Algorithm 1 yields an upper bound $v_1(s_{1l})$ for the value of the K-stage game, together with suboptimal strategies $\mathbf{f}_a$ and $\mathbf{g}_a$.*

The strategies $\mathbf{f}_a, \mathbf{g}_a$ of Algorithm 1 are possibly not pure. According to Theorem 3, we obtain $\tilde{v}_k(s_{kl}) \leqslant \bar{v}_k^p(s_{kl})$, and the proof holds for every $k = 2, \ldots, K$. Consider the value iteration for $k = 1$, with $\tilde{v}_2(s_{2l}) \leqslant \bar{v}_k^p(s_{2l})$, and $Q^{ij}(s_{1l}) = r^{ij}(s_{1l}) + \sum_{\delta_h \in S} P^{ij}(s_{2h}|s_{1l})$ $v_2^p(s_{2h}) \leqslant Q^{p,ij}(s_{2l})$, thus the true value of the K-stage game $v^*[Q(s_{1l})] \leqslant \bar{v}_1^p(s_{1l})$. The iterative value based on pure strategy auxiliary matrix sets $\mathbb{Q}_k^p$, $k = 1, \ldots, K$, obtained from Algorithm 1 is an upper bound for the game value. Let $v^*[Q(s_{na})]$ represent the minimum total payoff of the system when the strategy is calculated given that there is no attack at all in K stages, then $\bar{v}_1^p(s_{1l}) - v^*[Q(s_{1l})] \leqslant \bar{v}_1^p(s_{1l}) - v^*[Q(s_{na})]$, since $v^*[Q(s_{na})] \leqslant v^*[Q(s_{1l})]$ when the system operates in normal state without sacrificing any control cost to play against attacks. The sub-optimality of value $\bar{v}_1^p(s_{1l})$ calculated from Algorithm 1 is then bounded though we do not know the true value $v^*[Q(s_{1l})]$ of the game.

5. A moving-horizon approach for hybrid stochastic game

In this section, we propose a moving-horizon algorithm to compute the saddle-point equilibrium strategy at each stage of the hybrid stochastic game. A saddle-point equilibrium strategy is computed at each stage k by predicting anticipated future cost based on the hybrid state of the system $(x_{[k-T,k]}, \delta_l)$. We develop Algorithm 2 based on this concept, provide a scalable and a computationally tractable process, and compare the computational costs with Algorithm 1. The saddle-point equilibrium strategy and the value of the moving-horizon game at each stage involve solving finite zero-sum matrix games. By looking one stage ahead of the game state at k, predicting the physical dynamics $\mathbf{x}_{k+1}$ given any action pair, we obtain an objective function that reflects the payoff

of the current stage and future expectation for computing the strategies at k.

Given any action pair (a_{ik}, u_{jk}) at stage k, we first update the state space form of the system dynamics $\mathbf{x}_{k+1}$ based on $\mathbf{x}_{[k-T,k]}$ as (2). We view $\mathbf{x}_{k+1}$ as a function of $(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk})$, the immediate payoff function $r^{ij}(s_{(k+1)h})$ (for stage $k+1$) defined as (3) is also a function of the current game state and players' actions. We denote this relation as $r_{k+1}(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk}, \delta_h)$ in the following algorithms to distinguish it between definition (3), where the latter is the payoff results from the action of two players' at stage $k+1$. Then, we compute the value of the matrix game at stage $k+1$, by looking one stage ahead and consider stage $k+1$ as the terminal stage of the game, the value of game stage $k+1$ is now directly calculated via for $r^{ij}(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk}, \delta_h)$, $h = 1, 2, 3$, $i \in \{1, \ldots, M\}$, $j \in \{1, \ldots, N\}$ as (16):

$$v^{ij}_{k+1}(x_{[x-T,k]}, \delta_h) = \min_{\mathbf{g}} \max_{\mathbf{f}} (r(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk}, \delta_h)), \tag{16}$$

where $v_{k+1}(x_{[k-T,k]}, \delta_h) \in \mathbb{R}^{M\times N}$ is the value matrix of stage $k+1$ estimated at stage k based on the current game state and all possible action pairs. With the predicted value from the next stage, define the moving-horizon auxiliary matrix for stage k as:

$$\begin{aligned} &Q_k(s_{kl}) \\ &= r(s_{kl}) + \sum_{s_h \in S} P_k(s_{(k+1)h}|s_{kl}) \cdot v_{k+1}(x_{[k-T,k]}, \delta_h). \end{aligned} \tag{17}$$

The dot product of matrices $P_k(s_{(k+1)h}|s_{kl})$, $v_{k+1}(x_{[k-T,k]}, \delta_h)$ is an element-wise product of two elements at the same position of the two matrices. The value and stationary equilibrium strategies that Algorithm 2 calculates at each stage k are defined as follows.

Definition 5. Given s_{kl}, $v_{k+1}(x_{[k-T,k]}, \delta_h)$ as (16), and auxiliary matrix $Q_k(s_{kl})$ as (17), the value and equilibrium strategies at k are defined as the following equation:

$$v(s_{kl}) = \min_{\mathbf{g}_k(s_{kl})} \max_{\mathbf{f}_k(s_{kl})} \mathbf{f}_k(s_{kl})^T Q_k(s_{kl}) \mathbf{g}_k(s_{kl}), \tag{18}$$

where we treat the auxiliary matrix $Q_k(s_{kl})$ as the payoff matrix of a zero-sum game of stage k.

At each stage k, we repeat calculating $Q_k(s_{kl})$ and the corresponding value and equilibrium strategies, then update the system dynamics by the strategies for computation of next stage. The complete process is summarized as Algorithm 2.

Algorithm 2 (*Moving-Horizon Algorithm for A Hybrid Stochastic Game*)**.**
Input: System model parameters and game parameters.
Initialization: $\hat{\mathbf{x}}_{[0,T]}$.
Iteration: For $k = T, \ldots, K+T-1$, $s_{kl} = (x_{[k-T,k]}, \delta_l)$, $l = 1, 2, 3$: get the auxiliary matrix (17); compute the value and equilibrium strategies of every matrix game:

$$v(s_{kl}) = \min_{\mathbf{g}(s_{kl})} \max_{\mathbf{f}(s_{kl})} \mathbf{f}(s_{kl})^T Q_k(s_{kl}) \mathbf{g}(s_{kl}),$$

$$\mathbf{f}^*_k(s_{kl}) = \arg\max_{\mathbf{f}_k(s_{kl})} \mathbf{f}_k(s_{kl})^T Q_k(s_{kl}) \mathbf{g}^*_k(s_{kl}),$$

$$\mathbf{g}^*_k(s_{kl}) = \arg\min_{\mathbf{g}_k(s_{kl})} [\mathbf{f}^*_k(s_{kl})]^T Q_k(s_{kl}) \mathbf{g}_k(s_{kl}).$$

Update the system dynamics with strategies $\mathbf{f}^*_k(s_{kl})$, $\mathbf{g}^*_k(s_{kl})$, $l = 1, 2, 3$ as described in (2) for the next stage.
Return: the concatenation of strategies for both players $\mathbf{f} = \{f^*_k(s_{kl})\}$, $\mathbf{g} = \{\mathbf{g}^*_k(s_{kl})\}$ and the value sequence $v_k(s_{kl})$, $k = T, \ldots, K+T$, $l = 1, 2, 3$.

To get the total payoff till stage k by Algorithm 2, we plug the strategies $\mathbf{f}, \mathbf{g}$ into the system dynamics and calculate the sum of payoff for all stages. It is worth noting that Algorithm 2 reduces the computational overhead for the hybrid stochastic game. The complexity of Algorithm 2 is equivalent to the complexity of solving (KMN) times of minimax problem with an $M \times N$ payoff matrix, while the complexity of suboptimal Algorithm 1 is equivalent to the complexity of solving $((MN)^K)$ times of minimax problem with an $M \times N$ payoff matrix.

Remark 6. The system dynamics are defined by a sequence of action pairs (a_{ik}, u_{jk}) randomly chosen by the attacker and the system, and are equivalent with a system that randomly switches among N subsystems according to the stochastic game strategies $\mathbf{f}_k(s_{kl})$. The strategy sequences $\mathbf{f}^*_k(s_{kl})$, $\mathbf{g}^*_k(s_{kl})$ of the stochastic game converge to $\mathfrak{f}^l$, $\mathfrak{g}^l$, $l = 1, 2, 3$, i.e.,

$$\mathfrak{f}^l = \lim_{k\to\infty} \mathbf{f}^*_k(s_{kl}), \mathfrak{g}^l = \lim_{k\to\infty} \mathbf{g}^*_k(s_{kl}), l = 1, 2, 3,$$

if updating system dynamics at stage $k+1$ by $(\mathfrak{f}^l, \mathfrak{g}^l)$ results in:

$$\lim_{k\to\infty} Q_k(s_{kl}) = \lim_{k\to\infty} Q_k(s_{(k+1)l}), l = 1, 2, 3.$$

This is because according to Algorithm 2, $\mathbf{f}^*_k(s_{kl})$, $\mathbf{g}^*_k(s_{kl})$, $l = 1, 2, 3$ are the saddle-point equilibrium strategies for the auxiliary matrices $Q_k(s_{kl})$, $l = 1, 2, 3$. When the strategy sequences of both players converge, the switched system dynamics converge to a discrete-time Markov jump linear system (with delays when the attacker's strategies include replay attacks), and the stability properties of the system that switches among stable and unstable subsystems are analyzed by Zhang, Boukas, and Lam (2008) and Zhai, Hu, Yasuda, and Michel (2001).

6. Comparison of algorithms

One advantage of the moving horizon Algorithm 2 is its faster computation speed. Table 1 shows Matlab simulation time for different K-stage games, all with the same size of action space for the system and attacker. When K increases, the difference between algorithm speed also increases. We compare the cost of the strategies provided by the suboptimal Algorithms 1 and 2. The example studied is an unstable batch reactor, a four dimensional system (see Walsh, Ye, & Bushnell, 2002, Section IV.A for model parameters).

We first show the case under replay attacks, when the system is equipped with two controllers, one steady state Kalman filter, and the corresponding χ^2 detector. An optimal LQG controller u^*_K is denoted as controller 1, and a non-optimal controller $(u^*_k + \Delta u_k)$ (Mo & Sinopoli, 2009) with higher replay detection rate as controller 2. System's action space includes: subsystem u_{1k} with controller 1 and subsystem u_{2k} with controller 2. For illustration, we show the case when the attacker's action space are discretized replay attack time window size $\{10s, 20s, 30s, 40s\}$ in simulation. We design switched control policy for the system under replay attacks with initial mode δ_2, (i.e., $p(\delta^1_2) = 1$), we compare the system's strategies and total payoff when applying suboptimal strategies of Algorithm 1 and real-time receding horizon Algorithm 2 in a finite game of stage $K = 50$.

Fig. 3 shows the probability of switching to Controller 2 at every stage according to different algorithms. Three cases are shown in Fig. 4 — when the system applies the strategy of Algorithm 1, the strategy of Algorithm 2, and only the subsystem 2 with higher replay detection rate through all stages. Fig. 5 shows the probability that system being at mode δ_1 (successfully detected an attack), when applying strategies obtained from the two algorithms and always choosing subsystem 2. Applying a game strategy, randomly

Table 1
Elapsed time comparison of two algorithms.

K	Real time algorithm	Suboptimal algorithm
20	1.8054 s	6.7346 s
50	4.9968 s	58.6144s
100	8.3827 s	2073.2928 s
500	41.0342 s	20 h

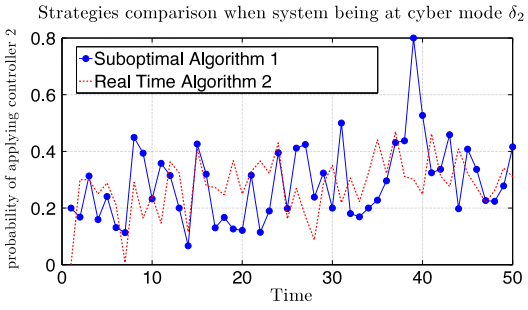


Fig. 3. Strategies comparison of two algorithms for system under replay attack—the probability of switching to subsystem 2 at mode δ_2 of every k.

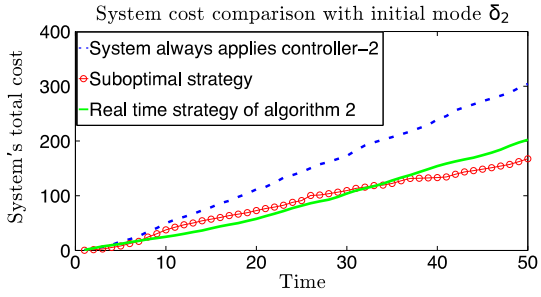


Fig. 4. Cost comparison of system applying different strategies at mode δ_2. Applying the suboptimal strategy provides the smallest cost, and the strategy of the real time algorithm is better than the one of a non-game approach.

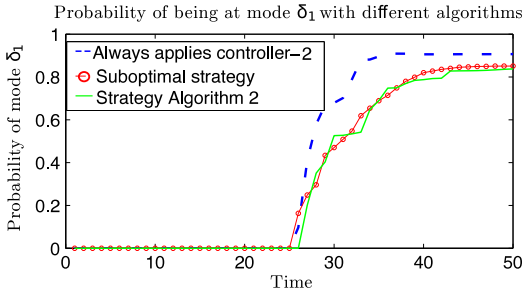


Fig. 5. Comparison of the probability of the system being at mode δ_1 for different strategies. Game strategies provide similar detection rate with the non-switching policy.

switching between subsystems results in a lower cost, while not sacrificing the detection rate significantly.

For game strategies designed for multiple types of attack, Fig. 6 shows the case when attacks are successfully detected and the system reaches the cyber mode $\delta_1 = safe$, the quadratic cost of the system converges. When replay finally occurs at $T_2 = 100$ s, with a game-theoretic strategy, the cost of the system is smaller than the cost when system always applies a controller with higher cost and higher detection rate. Data injection attacks shown in Fig. 6 appear during $k = 30, 31, \ldots, 50$.

These figures illustrate that the real-time strategy results a higher cost than the suboptimal system strategy, and they

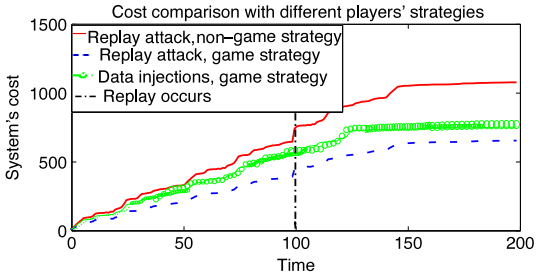


Fig. 6. Cost comparison when system and the attacker apply different strategies. The replay attack occurs at $T_2 = 100$ s.

both provide lower control costs compared to the non-game-theoretic approach. The non-game-theoretic approach provides only a slightly higher probability of being at the safe mode in K stages. By introducing the game strategy, i.e., switching between multiple subsystems, we do not sacrifice the payoff of the system while providing an acceptable detection rate, even we discretize the attacker's action space in the game framework. For instance, Figs. 4 and 5 show the result when the actual replay attack occurs at $T_2 = 25$ s, and the game strategies are calculated with action space $A_t = \{10s, 20s, 30s, 40s\}$. Since 25 is in between [20, 30], and the error of the action space discretization is bounded, a game strategy calculated via finite action space improves the system's performance.

7. Conclusion

In this work, we have proposed a zero-sum hybrid stochastic game model to capture the interactions between a cyber-physical system and an attacker—switching policy for the system under different types of sensor attacks. This framework allows us to find a control policy by calculating stationary strategy of the game with information of the system's physical dynamics and cyber modes. We design a suboptimal value iteration algorithm for a finite horizon game, which considers a saddle-point equilibrium of a robust stochastic game at each iteration. To reduce the computational complexity, a real-time moving-horizon algorithm is then developed. Based on the concept of saddle-point equilibrium for the hybrid stochastic game, at each stage, we look one stage ahead to calculate anticipated future value. The stability conditions of the system under multiple types of attacks based on the stochastic game framework, and an alternative algorithm with unknown transition matrix or payoffs will be our future work.

References

Aghassi, M., & Bertsimas, D. (2006). Robust game theory. *Mathematical Programming*, *107*(1), 231–273.

Basar, T., & Olsder, G. J. (1998). *Dynamic noncooperative game theory* (2nd ed.). Society for Industrial and Applied Mathematics.

Cardenas, A., Amin, S., Sionpoli, B., Perrig, A., & Sastry, S. (2009). Challenges for securing cyber physical systems. In *Workshop on future directions in cyber-physical systems security*. DHS.

Humphreys, T. (2013). Detection strategy for cryptographic GNSS anti-spoofing. *IEEE Transactions on Aerospace and Electronic Systems*, 1073–1090.

Kearns, M., Mansour, Y., & Singh, S. (2000). Fast planning in stochastic games. In *Proceedings of the 16th conference on uncertainty in artificial intelligence* (pp. 309–316).

Kim, K., & Kumar, P. (2012). Cyber-physical systems: A perspective at the centennial. *Proceedings of the IEEE*, *100*(Special Centennial Issue), 1287–1308.

Kroer, C., & Sandholm, T. (2015). Discretization of continuous action spaces in extensive-form games. In *Proceedings of the 2015 international conference on autonomous agents and multiagent systems* (pp. 47–56). Richland, SC.

Manshaei, M., Zhu, Q., Alpcan, T., Basar, T., & Hubaux, J. (2013). Game theory meets network security and privacy. *ACM Computing Surveys*, *45*(3), 25:1–25:39.

Miao, F., Pajic, M., & Pappas, G. J. (2013). Stochastic game approach for replay attack detection. In *53th IEEE conference on decision and control*.

Miao, F., & Zhu, Q. (2014). A moving-horizon hybrid stochastic game for secure control of cyber-physical systems. In *IEEE 53rd annual conference on decision and control* (pp. 517–522).

Miao, F., Zhu, Q., Pajic, M., & Pappas, G. (2016). Coding schemes for securing cyber-physical systems against stealthy data injection attacks. *IEEE Transactions on Control of Network Systems*, *4*(1), 106–117.

Mo, Y., Kim, T.-H., Brancik, K., Dickinson, D., Lee, H., Perrig, A., et al. (2012). Cyber-physical security of a smart grid infrastructure. *Proceedings of the IEEE*, *100*(1), 195–209.

Mo, Y., & Sinopoli, B. (2009). Secure control against replay attacks. In *47th annual allerton conference on communication, contro, and computing* (pp. 911–918).

Pajic, M., Tabuada, P., Lee, I., & Pappas, G. J. (2015). Attack-resilient state estimation in the presence of noise. In *2015 54th IEEE conference on decision and control* (pp. 5827–5832).

Pajic, M., Weimer, J., Bezzo, N., Tabuada, P., Sokolsky, O., & Lee, I. (2014) Robustness of attack-resilient state estimators. In *ACM/IEEE international conference on cyber-physical systems* (pp. 163–174).

Pasqualetti, F., Dorfler, F., & Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control*, *58*(11), 2715–2729.

Slay, J., & Miller, M. (2007). Lessons learned from the maroochy water breach. In *Critical infrast. protection* (pp. 73–82).

Walsh, G., Ye, H., & Bushnell, L. (2002). Stability analysis of networked control systems. *IEEE Transactions on Control Systems Technology*, *10*, 438–446.

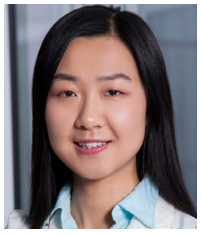
Zhai, G., Hu, B., Yasuda, K., & Michel, A. N. (2001). Stability analysis of switched systems with stable and unstable subsystems: An average dwell time approach. *International Journal of Systems Science*, *32*, 1055–1061.

Zhang, L., Boukas, E., & Lam, J. (2008). Analysis and synthesis of markov jump linear systems with time-varying delays and partially known transition probabilities. *IEEE Transactions on Automatic Control*, *53*(10), 2458–2464.

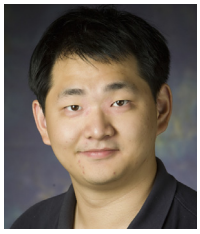
Zhong, M., Ding, X., Lam, J., & Wang, H. (2003). An LMI approach to design robust fault detection filter for uncertain LTI systems. *Automatica*, *39*(3), 543–550.

Zhu, M., & Martinez, S. (2011). Stackelberg-game analysis of correlated attacks in cyber-physical systems. In *American control conference, 2011* (pp. 4063–4068).

Zhu, Q., & Basar, T. (2015). Game-theoretic methods for robustness, security, and resilience of cyberphysical control systems: Games-in-games principle for optimal cross-layer resilient control systems. *IEEE Control Systems*, *35*(1), 46–65.



Fei Miao (S'13-M'16) is an Assistant Professor in the Department of Computer Science and Engineering at the University of Connecticut. Previously, she was a Postdoc Researcher in the Department of Electrical and Systems Engineering, GRASP Lab and PRECISE center, at the University of Pennsylvania. She received the B.Sc. degree in Automation from Shanghai Jiao Tong University in 2010, the M.A. degree in Statistics and the Ph.D. degree with the "Charles Hallac and Sarah Keil Wolf Award for Best Doctoral Dissertation" in Electrical and Systems Engineering from the University of Pennsylvania in 2015 and 2016, respectively. Her research interests include data-driven real-time optimization and control of cyber-physical systems under model uncertainties, and resilient and secure control cyber-physical systems. She was a Best Paper Award Finalist at the 6th ACM/IEEE International Conference on Cyber-Physical Systems in 2015.



Quanyan Zhu (S'04-M'12) is an assistant professor in the Department of Electrical and Computer Engineering at New York University. He received the B. Eng. in Honors Electrical Engineering with distinction from McGill University in 2006, the M.A.Sc. from University of Toronto in 2008, and the Ph.D. from the University of Illinois at Urbana-Champaign (UIUC) in 2013. From 2013–2014, he was a postdoctoral research associate at the Department of Electrical Engineering, Princeton University. He is a recipient of many awards including NSERC Canada Graduate Scholarship (CGS), Mavis Future Faculty Fellowships, and NSERC Postdoctoral Fellowship (PDF). He spearheaded and chaired INFOCOM Workshop on Communications and Control on Smart Energy Systems (CCSES), Midwest Workshop on Control and Game Theory (WCGT), and 7th Game and Decision Theory for Cyber Security (GameSec). His current research interests include resilient and secure interdependent critical infrastructures, energy systems, cyber-physical systems, and smart cities.



Miroslav Pajic (S'06-M'13) received the Dipl. Ing. and M.S. degrees in electrical engineering from the University of Belgrade, Serbia, in 2003 and 2007, respectively, and the M.S. and Ph.D. degrees in electrical engineering from the University of Pennsylvania, Philadelphia, in 2010 and 2012, respectively. He is currently an Assistant Professor in the Department of Electrical and Computer Engineering at Duke University. He also holds a secondary appointment in the Computer Science Department. Prior to joining Duke, Dr. Pajic was a Postdoctoral Researcher in the PRECISE Center, University of Pennsylvania, from 2012–2015. His research interests focus on the design and analysis of cyber-physical systems and in particular real-time and embedded systems, distributed/networked control systems, and high-confidence medical devices and systems. Dr. Pajic received various awards including ONR Young Investigator Award, NSF CAREER Award, the 2011 ACM SIGBED Frank Anger Memorial Award, the Best Paper Award at the 2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS), and the Best Student Paper award at the 2012 IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS).



George J. Pappas (S'90-M'91-SM'04-F'09) received the Ph.D. degree in electrical engineering and computer sciences from the University of California, Berkeley, CA, USA, in 1998. He is currently the Joseph Moore Professor and Chair of the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA. He also holds a secondary appointment with the Department of Computer and Information Sciences and the Department of Mechanical Engineering and Applied Mechanics. He is a Member of the GRASP Lab and the PRECISE Center. He had previously served as the Deputy Dean for Research with the School of Engineering and Applied Science. His research interests include control theory and, in particular, hybrid systems, embedded systems, cyber-physical systems, and hierarchical and distributed control systems, with applications to unmanned aerial vehicles, distributed robotics, green buildings, and bimolecular networks. Dr. Pappas has received various awards, such as the Antonio Ruberti Young Researcher Prize, the George S. Axelby Award, the Hugo Schuck Best Paper Award, the George H. Heilmeier Award, the National Science Foundation PECASE award and numerous best student papers awards at ACC, CDC, and ICCPS.