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

A Moving-Horizon Hybrid Stochastic Game for Secure Control of Cyber-Physical Systems

Fei Miao Quanyan Zhu

Abstract— **Security of cyber-physical systems (CPS) is a challenge for increasingly integrated systems today. To analyze and design detection and defense mechanisms for CPSs requires new system frameworks. In this paper, we establish a zero-sum hybrid stochastic game model, that can be used for designing defense policies for cyber-physical systems against attackers of different types. The hybrid game model contains physical states described by the system dynamics, and a cyber state that represents the detection mode of the system. A system selects a subsystem by combining one controller, one estimator and one detector among a finite set of candidate components at each state. In order to provide scalable and real-time computation of the switching strategies, we propose a moving-horizon approach to solve the zero-sum hybrid stochastic game, and obtain a saddle-point equilibrium policy for balancing the system's security overhead and control cost. This approach leads to a real-time algorithm that yields a sequence of Nash equilibrium strategies which can be shown to converge. The paper illustrates these concepts using numerical examples, and we compare the results with previously known designs.**

I. Introduction

Cyber Physical Systems (CPS) feature a tight integration of embedded computation, networks, and controlled physical processes [1]. The interaction among continuous physical dynamics, discrete communications, and computation substrates have made CPS vulnerable to malicious attacks beyond the standard cyber attacks [2]. Recoded attacks on CPS have brought into attention the challenges and requirements for secure CPS [1], [2], [3]. One famous incident, attack on Maroochy Water control system and the response discussed in [4], shows that CPS attacks can disrupt critical infrastructures and lead to undesirable, catastrophic consequences.

Xu *et al.* compare four different jamming attack models and detection schemes for consistency checking [5]. Syverson presents a taxonomy of replay attacks–independent of any analysis or preventing methods–on cryptographic protocols in [6]. In general, people use attack models as parameters to design defense schemes. However, a specific detection approach is not sufficient, when system is susceptible to various types of attacks and does not know which one will happen. Consequently, strategic methods that balance the system performance and security requirements are necessary, considering control and defense costs with the effects of multiple attacks.

The application of game theory to security problems has raised a lot of interest in recent years. Manshaei *et al.* summarize selected works that apply game-theoretic approaches in computer networks security and privacy problems [7]. Zhu *et al.* present a noncooperative stochastic game scheme of Intrusion Detection System (IDS) in [8]. A minimax game formulation in the presence of faults is discussed in [9]. Miao *et al.* design a zero-sum stochastic game approach for replay attack detection [10]. The game model parameters are quantified with the knowledge of system dynamics, and a suboptimal value iterative algorithm for finite-horizon non-stationary stochastic game is developed.

Building a scalable and computationally friendly framework 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 to capture the hybrid system dynamics and interactions with attacks. The hybrid game model contains a dynamic system model that captures the evolution of the physical processes, and discrete cyber modes that represent different security states of the CPS. We propose a computation methodology, that uses a moving window to select a sequence of physical state information, and computes a stationary saddle-point equilibrium strategy with the state being a joint cyber and physical state. This novel algorithm reduces the computational complexity of finding equilibrium solutions for the hybrid stochastic game, and yields an online algorithm for real-time CPS.

When the sequence of game strategies converges, the state transition probability of the game converges, and we leverage the stability analysis of Markov jump systems [11] to check system stability. The cost comparison with the suboptimal algorithm [10] 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 attacks.
2) We develop a real-time algorithm to reduce the computation overhead of the hybrid stochastic games, and analyze the convergence condition of the algorithm.

This paper is organized as follows. In Section II, we describe the system and attack models. In Section III, we formulate and quantify a zero-sum, hybrid stochastic game between the system and the attacker. The moving horizon

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.

F. Miao is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA, USA 19014. Q. Zhu is with Department of Electrical and Computer Engineering, New York University, Brooklyn, NY, USA 11201. Email: {miaofei,}@seas.upenn.edu,{quanyan.zhu}@nyu.edu

algorithm is shown, followed by an analysis of the algorithm convergence and system stability characteristics in Section IV. On several examples, in Section V we illustrate the computation speed and system performance of the derived algorithm. Finally, Section VI provides concluding remarks.

II. 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 constitute 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 Figure 1. We describe the model of each component in Figure 1 with a concrete example. The set of subsystems is not restricted to the models in the rest of this section, and the system model can be further generalized.

LTI Plant:: Consider a class of LTI plants described by:

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

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 $\mathbf{x}_0 \sim \mathcal{N}(\bar{\mathbf{x}}_0, \Sigma)$.

Estimators: Kalman filter is widely applied for noisy systems. We assume that $(\mathbf{A}, \mathbf{B})$ is stabilizable, $(\mathbf{A}, \mathbf{C})$ is detectable, then a steady state Kalman filter exists:

$$\begin{aligned} \hat{\mathbf{x}}_{0|-1} &= \bar{\mathbf{x}}_0, \mathbf{P}_{0|-1} = \Sigma, \mathbf{P}_{k+1|k} = \mathbf{A}\mathbf{P}_k\mathbf{A}^T + \mathbf{Q}, \\ \mathbf{P} &= \lim_{k\to\infty} \mathbf{P}_{k|k-1}, \mathbf{K} = \mathbf{P}\mathbf{C}^T(\mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R})^{-1}, \\ \hat{\mathbf{x}}_{k|k} &= \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(\mathbf{y}_k - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}), \\ \hat{\mathbf{x}}_{k+1|k} &= \mathbf{A}\hat{\mathbf{x}}_{k|k} + \mathbf{B}\mathbf{u}_k. \end{aligned} \quad (2)$$

Controllers: A state feedback control law is described as $\mathbf{u}_k = L(\hat{x}_{k|k})$, where $L(\cdot)$ is a linear function. For example, an optimal LQG controller is described as $\mathbf{u}_k = \mathbf{L}\hat{\mathbf{x}}_{k|k}$, where L is a time invariant matrix.

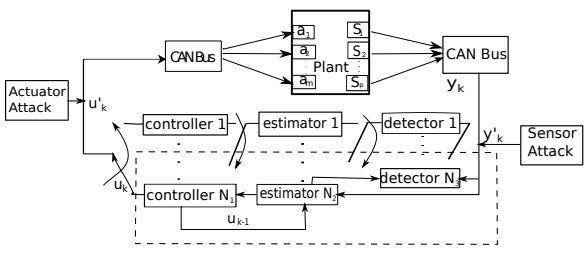

Fig. 1: Switching system diagram, where the system is equipped with N_1 controllers, N_2 estimators and N_3 detectors and switches among N subsystems. An example subsystem (controller N_1, estimator N_2, and detector N_3) is chosen in this figure.

Detectors: The detector design is related to the state estimator in general. For the steady state Kalman filter (2), the residues $\mathbf{z}_i = \mathbf{y}_i - \mathbf{C}\hat{\mathbf{x}}_{i|i-1}$ satisfying IID Gaussian $\mathcal{N}(0, \mathcal{P})$, where $\mathcal{P} = \mathbf{C}\mathbf{P}\mathbf{C}^T + \mathbf{R}$. Define the following

$$g_k = \sum_{i=k-\tau+1}^{k} (\mathbf{y}_i - \mathbf{C}\hat{\mathbf{x}}_{i|i-1})^T \mathcal{P}^{-1} (\mathbf{y}_i - \mathbf{C}\hat{\mathbf{x}}_{i|i-1}),$$

and g_k satisfies χ^2 distribution. A χ^2 square detector triggers the alarm when $g_k \geq \alpha$. We assume that the detection window size τ and the threshold α are provided as system model according to an expected false alarm trigger rate.

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 $\delta_1 = safe$ describes that the system has already successfully detected an attack; $\delta_2 = no\ detection$ specifies that the alarm is not triggered; finally, the system enters the state $\delta_3 = false\ alarm\ trigger$ when the alarm is triggered while no attack has yet occurred. The mode depends on the detector, and is a probability information. We assume that once the alarm is triggered, the system will stop the execution and check whether some attack occurred or it is a false alarm trigger; when the system is hijacked, the estimator, detector and controller are fed with false data, until an alarm is triggered and the system reacts to the attack.

Attack model: We assume that the controller/estimator/detector are secured, and attackers can not hack code implemented in these components. Sensors and actuators are vulnerable, and attacker can change values sent from sensors or received by actuators – $\mathbf{y}_k, \mathbf{u}_k$ of system (1) are defined as $\mathbf{y}'_k, \mathbf{u}'_k$, according to the types of attacks we consider. For example, data injection attacks change the vectors as: $\mathbf{y}'_k = \mathbf{y}_k + \mathbf{y}^a_k, \mathbf{u}'_k = \mathbf{u}_k + \mathbf{u}^a_k$; replay attacks change sensor values as $\mathbf{y}'_k = \mathbf{y}_{k-T_2}$, where T_2 is the replay window size.

III. 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 joint game state space $(X_{[k-T,k]} \times S)$ contains information about both the system dynamics $\mathbf{x}_k$ and the discrete modes $\delta_l, l = 1, 2, 3$. With this game state definition, the state transition between stage k and $k+1$ is Markov – the joint state includes information we need to compute the game strategy at the current stage. At each stage $k \in \{T, \cdots, K+T\}$, other game parameters include action space for the attacker (system) A_{tk} (A_{sk}), the state transition probability matrix $\mathbb{P}_k$, and the immediate payoff matrix r_k. The solution set of the game are 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: $\{(X_{[k-T,k]} \times S), A_{tk}, A_{sk}, \mathbf{F}_k, \mathbf{G}_k, \mathbb{P}_k, r_k\}$.

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

$x_{[k-T,k]} = (x_{k-T}, x_{k-T+1}, \cdots, x_k) \in X_{[k-T,k]}$ is the discrete time dynamics of the physical process provided to the system–the state estimations $\hat{x}_{k-T}, \cdots, \hat{x}_k$, $\delta_l \in S = \{\delta_1, \delta_2, \delta_3\}$ denote the cyber state of the system. We assume that once the game reach 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 Figure 2. Here T is the window size of system dynamics that we need to quantify the parameters at game stage k. For example, if the χ^2 detector's detection window size is T_1, considering sensor 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: For definition simplicity, we consider sensor attacks $\mathbf{y}'_k \in A_{tk}$, where $A_{tk} = \{a_{1k}, a_{2k}, \cdots, a_{Mk}\}$ is the attacker's action space at stage k, and a_{1k} means no attack. The actions can be either multiple types, or the same type attack with different values. [1]

System's Action Space: $A_{sk} = \{u_{1k}, u_{2k}, \cdots, u_{Nk}\}$ is the system's action space at stage k, where u_{jk} is the index for the jth subsystem. We assume that the N subsystems (a model for each component in Figure 1) are pre-determined. For example, a subsystem can be the plant with a given optimal LQG controller, a Kalman filter and a χ^2 detector.

Mixed Strategy: Let $f^i_k(s_{kl})$ ($g^j_k(s_{kl})$) be the probability that the attacker (system) chooses action $a_{ik} \in A_{tk}$ ($u_{jk} \in A_{sk}$) at state $s_{kl} \in (X_{[k-T,k]} \times S)$. Define $\mathbf{F}_k$ and $\mathbf{G}_k$ as the strategy sets of the attacker and the system for stage k:

$$\mathbf{F}_k := \{\mathbf{f}_k = [\mathbf{f}_k(s_{k1}), \mathbf{f}_k(s_{k2}), \mathbf{f}_k(s_{k3})] | f^i_k(s_{kl}) \geq 0, \sum_{a_{ik} \in A_{tk}} f^i_k(s_{kl}) = 1, \mathbf{f}_k(s_{kl}) \in \mathbb{R}^M, \forall s_{kl} \in (X_{[k-T,k]}, S)\},$$

$$\mathbf{G}_k := \{\mathbf{g}_k = [\mathbf{g}_k(s_{k1}), \mathbf{g}_k(s_{k2}), \mathbf{g}_k(s_{k3})] | g^j_k(s_{kl}) \geq 0, \sum_{u_{jk} \in A_{sk}} g^j_k(s_{kl}) = 1, \mathbf{g}_k(s_{kl}) \in \mathbb{R}^N, \forall s_{kl} \in (X_{[k-T,k]}, S)\}.$$

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.

[1] To find system's strategy based on the game formulation, it is required that attacker's action space is defined. If the attacker's actual behavior is outside of the action space we consider, then a switched system does not ensure performance under the attack outside the action space.

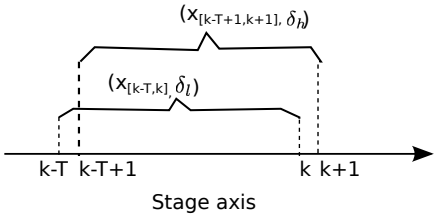

Fig. 2: Joint state transition of the hybrid stochastic game when moving the horizon of game state one step ahead. When the state transits from stage k to $k+1$, we slice the window of the sequence of physical dynamics one step ahead, add x_{k+1} and remove x_{k-T}, thus $x_{[k-T,k]} \to x_{[k-T+1,k+1]}$.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

System Dynamic under game framework: Given the subsystem and attack models in Section II and the game definition, we show the dynamics at stage k given an action pair (a_{ik}, u_{jk}) (assume initial $\hat{\mathbf{x}}_{1|0} = \bar{\mathbf{x}}_0$, $\mathbf{x}_1 = \mathbf{x}_0$).

A subsystem: Each action pair (a_{ik}, u_{jk}) defines the corresponding system dynamics at k. For example, when we focus on sensor attacks (like replay or false data injection), let $\gamma_k(a_{ik}, u_{jk})$ be the control input with (a_{ik}, u_{jk}), a subsystem u_{jk} with a Kalman filter, an optimal LQG controller, and a χ^2 detector, has the following dynamics:

$$\begin{aligned}
&\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} + \mathbf{B}\mathbf{u}_{k-1} + \mathbf{w}_{k-1},\\
&\mathbf{y}_k = \begin{cases} a_{1k} = \mathbf{C}\mathbf{x}_k + \mathbf{v}_k, & \text{without attack} \\ a_{ik}, i = 2, \cdots, M, & \text{with attack} \end{cases}\\
&\hat{\mathbf{x}}_{k|k-1} = \mathbf{A}\hat{\mathbf{x}}_{k-1|k-1} + \mathbf{B}\mathbf{u}_{k-1},\\
&\hat{\mathbf{x}}_{k|k}(a_{ik}) = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}(a_{ik} - \mathbf{C}\hat{\mathbf{x}}_{k|k-1}),\\
&\hat{\mathbf{x}}_{k+1|k}(a_{ik}, u_{jk}) = \mathbf{A}\hat{\mathbf{x}}_{k|k}(a_{ik}) + \mathbf{B}\gamma_k(a_{ik}, u_{jk}),\\
&\gamma_k(a_{ik}, u_{jk}) = \mathbf{L}\hat{\mathbf{x}}_{k|k}(a_{ik}),\\
&\mathbf{z}_{k+1}(a_{ik}, u_{jk}) = a_{ik} - \mathbf{C}\hat{\mathbf{x}}_{k+1|k}(a_{ik}, u_{jk}).
\end{aligned} \tag{3}$$

Consequently, the sum of residues g_{k+1} and the probability of triggering an alarm by the χ^2 detector at $k+1$ are:

$$\begin{aligned}
g_{k+1}(a_{ik}, u_{jk}) &= \sum_{t=k-\tau+1}^{k} [\mathbf{z}_t]^T \mathcal{P}^{-1} \mathbf{z}_t\\
&+ [\mathbf{z}_{k+1}(a_{ik}, u_{jk})]^T \mathcal{P}^{-1} \mathbf{z}_{k+1}(a_{ik}, u_{jk}),\\
P(g_{k+1}(a_{ik}, u_{jk})) &\geq \alpha.
\end{aligned} \tag{4}$$

State Transition Probability: Given a set of subsystem models, define the state transition probability as $(X_{[k-T,k]} \times S) \times A_{tk} \times A_{sk} \to P(X_{[k-T+1,k+1]} \times S)$, where

$$\begin{aligned}
&\tilde{P}_k(s_{(k+1)h}|s_{kl}) = [\tilde{P}^{ij}_k(s_{(k+1)h}|s_{kl}) \geq 0] \in \mathbb{R}^{M \times 2},\\
&s_{(k+1)h} \in (X_{[k-T+1,k+1]} \times S), s_{kl} \in (X_{[k-T,k]} \times S),\\
&\sum_{s_{(k+1)h} \in (X_{k+1} \times S)} \tilde{P}^{ij}_k(s_{(k+1)h}|s_{kl}) = 1,\\
&\forall (a_{ik}, u_{jk}) \in A_{tk} \times A_{sk}, s_{kl} \in (X_{[k-T,k]} \times S).
\end{aligned}$$

$\tilde{P}^{ij}_k(s_{(k+1)h}|s_{kl})$ is the probability that system transit 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. The transition probability is provided by intrusion detectors of the subsystem. For example, if a χ^2 detector is the detector component of subsystem u_{jk}, we apply (4) to decide the state transition probability.

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}). Define $r_k : (X_{[k-T,k]} \times S) \times A_{tk} \times A_{sk} \to \mathbb{R}$, where $\tilde{r}_k(s_{kl}) = [\tilde{r}^{ij}_k(s_{kl}) \geq 0]$. Let $\gamma_k(a_{ik}, u_{jk})$ be the control input given action pair (a_{ik}, u_{jk}). For example, considering expected linear quadratic cost

$$\begin{aligned}
\tilde{r}^{ij}_k(s_{kl}) =& \tilde{r}^{ij}_{tk}(s_{kl}) = -\tilde{r}^{ij}_{sk}(s_{kl}), \text{ and define:}\\
\tilde{r}^{ij}_k(s_{k1}) =& \mathbb{E}\hat{\mathbf{x}}^T_k \mathbf{W} \mathbb{E}\hat{\mathbf{x}}_k + \mathbb{E}\gamma^T_k(a_{1k}, u_{jk}) \mathbf{U} \mathbb{E}\gamma_k(a_{1k}, u_{jk}),\\
\tilde{r}^{ij}_k(s_{k2}) =& \mathbb{E}\hat{\mathbf{x}}^T_k \mathbf{W} \mathbb{E}\hat{\mathbf{x}}_k + \mathbb{E}\gamma^T_k(a_{ik}, u_{jk}) \mathbf{U} \mathbb{E}\gamma_k(a_{ik}, u_{jk}),\\
\tilde{r}^{ij}_k(s_{k3}) =& p_f,
\end{aligned} \tag{5}$$

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

Expected Model Update With Strategy at Stage k: Let $p(s_{kl})$ be the probability system is at state s_{kl} at stage k ($p(s_{1l})$ 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:

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

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

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

IV. A Moving-Horizon Approach for Hybrid Stochastic Game

In this section, we propose a moving-horizon algorithm to compute the saddle-point equilibrium strategy of the hybrid stochastic game. Illustrated in Fig. 2, a time window of size T is used, and an equilibrium strategy is computed at each stage k by looking back T stages of the physical state $x_{[k-T,k]}$ and its associated cyber state δ_l. Detailed process of moving the horizon to obtain predicted future stage information is described in Subsection A. Algorithm 1 is developed based on this concept, and provides a scalable and real-time computation process, which allows us to analyze the convergence property of the strategies of the hybrid stochastic game in Subsection B.

A. A Moving-Horizon Algorithm for Game Strategies

The saddle-point equilibrium strategy and the value of the moving-horizon game at each stage involves solving finite zero-sum matrix games. In this paper, we consider an objective function that reflects the payoff of the game at the current stage k, and also the expected payoff from the future stage. By looking one stage ahead of the game state at k, predicting the physical dynamics $\mathbf{x}_{k+1}$ given any action pair, we move the information horizon to stage $k+1$ and obtain future expectation for computing the strategies at stage k. The moving horizon process is illustrated as Figure 2. Detailed process to construct the payoff matrix of a zero-sum game for stage k is described, and Algorithm 1 presents the complete equilibrium computation process of the hybrid stochastic game.

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 (3). We view $\mathbf{x}_{k+1}$ as a function of $(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk})$, the immediate payoff function $\tilde{r}_{k+1}(s_{(k+1)h})$ (for stage $k+1$) defined as (5) is a function of $s_{(k+1)h} = (\mathbf{x}_{[k-T+1,k+1]}, \delta_h)$, thus r_{k+1} is a function of $(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk}, \delta_h)$, as shown in the following equation (6). Then, we compute the value of the matrix game at stage $k+1$, for every $r_{k+1}(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk}, \delta_h)$, $h = 1, 2, 3$, $i \in \{1, \cdots, M\}, j \in \{1, \cdots, N\}$ as (6):

$$v_{k+1}^{ij}(x_{[x-T,k]}, \delta_h) = \min_{\mathbf{g}} \max_{\mathbf{f}} (r_{k+1}(\mathbf{x}_{[k-T,k]}, a_{ik}, u_{jk}, \delta_h)) \tag{6}$$

With the predicted value from the next stage, define the auxiliary matrix for stage k as:

$$Q_k(s_{kl}) = r_k(s_{kl}) + \sum_{s_h \in S} \tilde{P}_k(s_{(k+1)h}|s_{kl}) \cdot v_{k+1}(x_{[k-T,k]}, \delta_h), \tag{7}$$

where the matrix $v_{k+1}^{ij}(x_{[x-T,k]}, \delta_h)$ is defined by (6), and it is the element of the ith row, jth column of the matrix

$$v_{k+1}(x_{[k-T,k]}, \delta_h) \in \mathbb{R}^{M \times N}.$$

The dot products between two matrices $\tilde{P}_k(s_{(k+1)h}|s_{kl})$, $v_{k+1}(x_{[k-T,k]}, \delta_h)$ is an element wise product of two elements at the same position of the two matrices.

The value and stationary equilibrium strategies that Algorithm 1 calculates at each stage k is defined as following:

Definition 1: Given s_{kl}, $v_{k+1}(x_{[k-T,k]}, \delta_h)$ as (6), and auxiliary matrix $Q_k(s_{kl})$ as (7), the value and equilibrium strategies at k are defined as the following equation:

$$v(s_{kl}) = \min_{\mathbf{g}_k(s_{kl})} \max_{\mathbf{f}_k(s_{kl})} \mathbf{f}_k(s_{kl})^T Q_k(s_{kl}) \mathbf{g}_k(s_{kl}). \tag{8}$$

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 1. To get the total payoff till stage k by Algorithm 1, we plug in the strategies $\mathbf{f}, \mathbf{g}$ to the system dynamics and calculate the sum of payoff for all stages.

Remark 1: It is worth noting that Algorithm 1 reduces the computation overhead for the hybrid stochastic game, since it looks one stage ahead with a moving-horizon information window. The complexity of Algorithm (1) is $O(K)$. For a large total stage number of the hybrid stochastic game $\tilde{T}$, it is necessary to examine the strategy trend of Algorithm 1, such as convergence property. As a contrast, the suboptimal algorithm in [10] takes the total expected payoff as an objective function. The complexity of suboptimal algorithm is exponential with stage number K, because the algorithm looks K stages ahead at once and compute a robust game for every iteration. The advantage of suboptimal algorithm in [10] is to provide an upper bound of the total finite cost. However, for a large $\tilde{T}$, the suboptimal algorithm in [10] is computationally expensive. Numerical comparisons are shown in Section V.

□

Algorithm 1 : Moving-Horizon Algorithm for A Hybrid Stochastic Game

Input: System model parameters and game parameters.

Initialization: $\hat{\mathbf{x}}_{1|0}, \mathbf{x}_1$.

Iteration: For $k = T, \cdots, K+T-1$, $s_{kl} = (x_{[k-T,k]}, \delta_l)$, $l = 1, 2, 3$: get the auxiliary matrix (7) for stage k; 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 3 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 $v_k(s_{kl}), k = 1, \cdots, K, l = 1, 2, 3$.

B. Convergence Analysis of the Algorithm

Given the sets of models for each component of the subsystems and attacks, the system dynamics are defined by a sequence of action pairs $(a_{ik}, u_{jk}), k \in \{k+T, \cdots, K+T\}$ randomly chosen by the attacker and the system. Then, the system dynamics with the stochastic game strategies (for the system and the attacker) are equivalent with a switched system – the system model randomly switches among N subsystems, according to strategies $\mathbf{f}_k(s_{kl})$. The following theorem shows the existence condition of convergent strategies when $k \to \infty$. When there exists such strategies, the switched system can be described as a Markov jump system, since the state transition probability also converges.

Proposition 1: 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:

$$\lim_{k\to\infty} Q_k(s_{kl}) = \lim_{k\to\infty} Q_k(s_{(k+1)l}), l = 1, 2, 3. \quad (9)$$

Proof: According to Algorithm 1, the strategies $\mathbf{f}_k^*(s_{kl})$, $\mathbf{g}_k^*(s_{kl}), l = 1, 2, 3$ are the saddle-point equilibrium strategies for the payoff matrices $Q_k(s_{kl}), l = 1, 2, 3$. Thus, if (9) holds, the auxiliary matrix $Q_k(s_{kl})$ converges, and we get convergent strategies for both players. ■

Remark 2: 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), then we analyze the stability properties of the system based on conclusions of previous work [11]. □

It is possible that some subsystems $u_{jk}, j \in \{1, \cdots, N\}$ are unstable under specific types of attacks. When this is the case, the system switches among stable and unstable subsystems. Stability properties of continuous time linear switched systems including unstable modes are analyzed in [12]. To guarantee exponential stability, the total activation time of unstable subsystems need to be relatively small compared with that of stable subsystems. Given the stochastic game strategy, we get the switched dynamic process of the system under different types of attacks, and check whether stability conditions are violated. More analysis of system stability conditions based on the moving horizon stochastic game framework will be an avenue of future work.

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

TABLE I: Elapsed time comparison of two algorithms

V. Comparison of Algorithms

One advantage of the moving horizon Algorithm 1 is its faster computation speed. Table I shows Matlab simulation time for different K-stage games, all with a (4×2) action space (i.e., the attacker has 4 actions and the system has 2 actions). When K increases, the difference between algorithm speed also increases.

Using a linear system with control-cost optimal (but nonsecure) and secure (but cost-suboptimal) controllers in presence of replay attacks as an example, we compare the cost of the strategies provided by the suboptimal algorithm in [10] and Algorithm 1. The example studied is an unstable batch reactor [13], which is a four dimensional system. The linearized model parameters are:

$$\mathbf{A} = \begin{bmatrix} 1.38 & -0.2077 & 6.715 & -5.676 \\ -0.5814 & -4.29 & 0 & 0.675 \\ 1.067 & 4.273 & -6.654 & 5.893 \\ 0.048 & 4.273 & 1.343 & -2.104 \end{bmatrix},$$

$$\mathbf{B} = \begin{bmatrix} 0 & 0 \\ 5.679 & 0 \\ 1.136 & -3.14 \\ 1.136 & 0 \end{bmatrix}, \mathbf{C} = \begin{bmatrix} 1 & 0 & 1 & -1 \\ 0 & 1 & 0 & 0 \end{bmatrix}, \mathbf{D} = \mathbf{0}$$

With the system parameters, we compute two controllers: Controller 1 is the optimal LQG controller u_K^*; controller 2 is $u_k^* + \Delta u_k$, a non-optimal controller with higher replay detection rate as designed by [14]. The system has one steady state Kalman filter, and the corresponding χ^2 detector. Assume that there are two subsystems: subsystem u_{1k} is with controller 1, and subsystem u_{2k} is with controller 2. The replay attack window sizes (attacker's action space) are $\{10s, 20s, 30s, 40s\}$ and we design switched control policy for the system. We assume that the initial system mode is δ_2, (i.e., $p(\delta_2^1) = 1$), the total stage number $K = 50$.

Figure 3 shows the probability of switching to Controller 2 at every stage according to different algorithms. Three cases are shown in Figure 4–when the system applies the strategy of Algorithm 1 in this work, the suboptimal algorithm strategy and only the cost non-optimal controller through all stages. Figures 5 shows the probability that system being at mode δ_1 (successfully detected an attack), when applying strategies obtained from the two algorithms and

only applying the controller providing a higher detection rate for replay attack. Applying a game strategy, randomly switching between subsystems results in a lower cost, while does not sacrifice the detection rate much. The suboptimal algorithm performs better with respect to cost saving.

Game strategies still provide system performance improvement compared with a non-game approach, even only the attack type is included in the attacker's action space of the game framework but not the exact behavior of the attacker. For example, consider a replay attack $T_2 = 25s$, and the game strategy calculated with action space $\{10, 20, 30, 40\}$. Since 25 is in the range of $[20, 30]$, the payoff and state transition probability are approximated by the parameters when T_2 is $20, 30$.

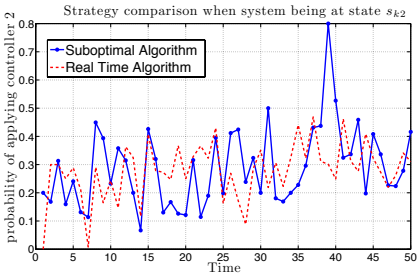

Fig. 3: Strategies comparison of two algorithms for system under replay attack–the probability of switching to subsystem 2 at mode δ_2 of every k.

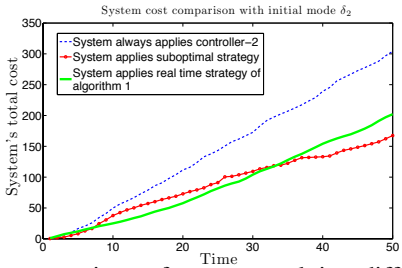

Fig. 4: Cost comparison of system applying different strategies at mode δ_2. Applying the suboptimal strategy provides the smallest cost, and the strategy of the real time algorithm is better than the one of a non-game approach.

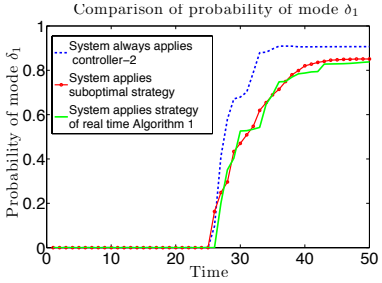

Fig. 5: Comparison of the probability of the system being at mode δ_1 for different strategies. Game strategies provide similar detection rate with the non-switching policy.

VI. 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 active attacks. To reduce the computational complexity, a real-time algorithm is developed based on the concept of moving-horizon computation of saddle-point equilibrium for the hybrid stochastic game framework. At each step, we look ahead one stage, with information of a window of physical dynamics and cyber modes, to compute an equilibrium policy. In the future, we plan to analyze stability conditions of the system based on the stochastic game framework.

Acknowledgements

The author would like to thank Miroslav Pajic, George J. Pappas, both from University of Pennsylvania for fruitful discussions about the problem and helpful comments.

REFERENCES

[1] K.-D. Kim and P. R. Kumar, "Cyber-physical systems: A perspective at the centennial," *Proceedings of the IEEE*, vol. 100, no. special Centennial-Issue, 2012.

[2] A. Cardenas, S. Amin, and S. S. Sastry, "Research challenges for the security of control systems." in *Proceedings of the 3rd USENIX Workshop on Hot topics in security*, 2008, Article 6.

[3] A. Cardenas, S. Amin, B. Sionpoli, A. Perrig, and S. Sastry, "Challenges for securing cyber physical systems," Workshop on future directions in cyber-physical systems security, 2009.

[4] J. Slay and M. Miller, "Lessons learned from the maroochy water breach," in *Critical Infrast. Protection*, 2007, pp. 73–82.

[5] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, 2005, pp. 46–57.

[6] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *Computer Security Foundations Workshop VII*, 1994, pp. 187–191.

[7] M. Manshaei, Q. Zhu, T. Alpcan, T. Basar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM Comput. Surv.*, vol. 45, no. 3, pp. 25:1–25:39, 2013.

[8] Q. Zhu and T. Basar, "Dynamic policy-based ids configuration," in *48th IEEE Conference on Decision and Control (CDC)*, 2009, pp. 8600–8605.

[9] S. Verdu and H. Poor, "On minimax robustness: A general approach and applications," *IEEE Transactions on Information Theory*, vol. 30, no. 2, pp. 328–340, 1984.

[10] F. Miao, M. Pajic, and G. J. Pappas, "Stochastic game approach for replay attack detection," in *53th IEEE Conference on Decision and Control*, 2013.

[11] L.Zhang, E.K.Boukas, and J.Lam, "Analysis and synthesis of markov jump linear systems with time-varying delays and partially known transition probabilities," *IEEE Transactions on Automatic Control*, vol. 53, no. 10, pp. 2458–2464, 2008.

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

[13] G. Walsh, H. Ye, and L. Bushnell, "Stability analysis of networked control systems," *IEEE Transactions on Control Systems Technology*, vol. 10, pp. 438–446, 2002.

[14] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *47th Annual Allerton Conference on Communication, Contro, and Computing*, 2009, pp. 911–918.