

A dynamic games approach to proactive defense strategies against Advanced Persistent Threats in cyber-physical systems

Linan Huang*, Quanyan Zhu

Department of Electrical and Computer Engineering, New York University, Brooklyn, NY 11201, USA



ARTICLE INFO

Article history:

Received 4 May 2019

Revised 5 October 2019

Accepted 4 November 2019

Available online 6 November 2019

Keywords:

Advanced persistent threats

Defense in depth

Proactive defense

Industrial control system security

Cyber deception

Multi-stage Bayesian game

Perfect Bayesian Nash equilibrium

Tennessee Eastman process

ABSTRACT

Advanced Persistent Threats (APTs) have recently emerged as a significant security challenge for a cyber-physical system due to their stealthy, dynamic and adaptive nature. Proactive dynamic defenses provide a strategic and holistic security mechanism to increase the costs of attacks and mitigate the risks. This work proposes a dynamic game framework to model a long-term interaction between a stealthy attacker and a proactive defender. The stealthy and deceptive behaviors are captured by the multi-stage game of incomplete information, where each player has his own private information unknown to the other. Both players act strategically according to their beliefs which are formed by the multi-stage observation and learning. The perfect Bayesian Nash equilibrium provides a useful prediction of both players' policies because no players benefit from unilateral deviations from the equilibrium. We propose an iterative algorithm to compute the perfect Bayesian Nash equilibrium and use the Tennessee Eastman process as a benchmark case study. Our numerical experiment corroborates the analytical results and provides further insights into the design of proactive defense-in-depth strategies.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

The recent advances in automation technologies, 5G networks, and cloud services have accelerated the development of cyber-physical systems (CPSs) by integrating computing and communication functionalities with components in the physical world. Cyber integration increases the operational efficiency of the physical system, yet it also creates additional security vulnerabilities. First, the increased connectivity and openness have expanded the attack surface and enabled attackers to leverage vulnerabilities from multiple system components to launch a sequence of stealthy attacks. Second, the component heterogeneity, the functionality complexity, and the dimensionality of cyber-physical systems have created many zero-day vulnerabilities, which make the defense arduous and costly.

Advanced Persistent Threats (APTs) are a class of emerging threats for cyber-physical systems with the following distinct features. Unlike opportunistic attackers who spray and pray, APTs have specific targets and sufficient knowledge of the system architecture, valuable assets, and even defense strategies. Attackers can tailor their strategies and invalidate cryptography, firewalls, and in-

trusion detection systems. Unlike myopic attackers who smash and grab, APTs are stealthy and can disguise themselves as legitimate users for a long sojourn in the victim's system.

A few security researchers and experts have proposed APT models in which the entire intrusion process is divided into a sequence of phases, such as Lockheed-Martin's Cyber Kill Chain (see Hutchins et al., 2011), MITRE's ATT&CK (see Corporation, 2019), the NSA/CSS technical cyber threat framework (see Department of Homeland Security, 2018), and the ones surveyed in Messaoud et al. (2016). Fig. 1 illustrates the multi-stage structure of APTs. During the reconnaissance phase, a threat actor collects open-source or internal intelligence to identify valuable targets. After the attacker obtains a private key and establishes a foothold, he escalates privilege, propagates laterally in the cyber network, and eventually either accesses confidential information or inflicts physical damage. Static standalone defense on a physical system cannot deter attacks originated from a cyber network.

The multi-phase feature of APTs results in the concept of Defense in Depth (DiD), i.e., multi-stage cross-layer defense policies. A system defender should adopt defensive countermeasures across the phases of APTs and holistically consider interconnections and interdependencies among these layers. To formally describe the interaction between an APT attacker and a defender with the defense-in-depth strategy, we map the sequential phases of APTs into a game of multiple stages. Each stage describes a local interaction between the attacker and the defender where the

* Corresponding author.

E-mail addresses: lh2328@nyu.edu (L. Huang), qz494@nyu.edu (Q. Zhu).

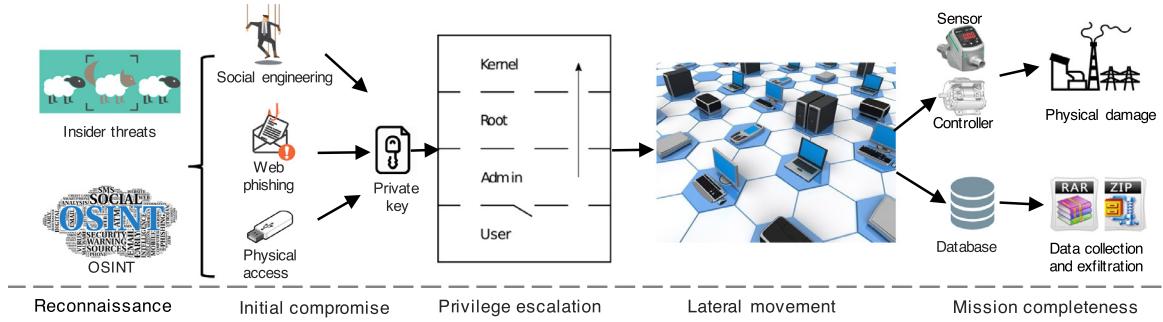


Fig. 1. An illustrate example of the multi-stage structure of APTs. The multi-stage attack is composed of reconnaissance, initial compromise, privilege escalation, lateral movement, and mission execution. An attack originated from an early-stage cyber network can lead to damage in a physical system.

outcome leads to the next stage of interactions. The goal of the attacker is to stealthily reach the targeted physical or informational assets while the defender aims to take defensive actions at multiple phases to thwart the attack or reduce its impact.

Detecting APTs timely (i.e., before attackers have reached the final stage) and effectively (i.e., with a low rate of false alarms and missed detections) is still an open problem due to their stealthy and deceptive characteristics. As reported in LLC (2018), US companies in 2018 have taken an average of 197 and 69 days, respectively, to detect and contain a data breach. Stuxnet-like APT attacks can conceal themselves in a critical industrial system for years and inconspicuously increase the failure probability of physical components. Due to the insufficiency of timely and effective detection systems for APTs, the defender remains uncertain about the user's type, i.e., either legitimate or adversarial, throughout stages. To prepare for the potential APT attacks, the defender needs to adopt precautions and proactive defense measures, which may also impair the user experience and reduce the utility of a legitimate user. Therefore, the defender needs to strategically balance the tradeoff between security and usability when the user's type remains private.

In this work, we model the private information of the user's type as a random variable following the work of Harsanyi (1967). Under the same defense action, the behavior and the utility of a user depend on whether his type is legitimate or adversarial. To make secure and usable decisions under incomplete information, the defender forms a belief on the user's type and updates the belief via the Bayesian rule based on the information acquired at each stage. For example, throughout the phases of an APT, detection systems can generate many alerts based on suspicious user activities. Although these alerts do not directly reveal the user's type, a defender can use them to reduce the uncertainty on the user's type and better determine her defense-in-depth strategies at multiple stages.

Defensive deception provides an alternative perspective to bring uncertainty to the attacker and tilt the information asymmetry. We classify a defender into different levels of sophistication based on factors such as her level of security awareness, detection techniques she have adopted, and the completeness of her virus signature database. A sophisticated defender has a higher success rate of detecting adversarial behaviors. Thus, the behavior of an attacker depends on the type of defender that he interacts with. For example, the attacker may remain stealthy when he interacts with a sophisticated defender but behaves more aggressively when interacting with a primitive defender. As the attacker has incomplete information regarding the defender's type, he needs to form a belief and continuously updates it based on his observation of the defender's actions. In this way, the attacker can optimally decide whether, when, and to what extent, to behave aggressively or conservatively.

To this end, we also use a random variable to characterize the private information of the defender's type. As both players have incomplete information regarding the other player's type and they make sequential decisions across multiple stages, we extend the classical static Bayesian game to a multi-stage nonzero-sum game with two-sided incomplete information. Both players act strategically according to their beliefs to maximize their utilities. The Perfect Bayesian Nash Equilibrium (PBNE) provides a useful prediction of their policies at every stage for each type since no players can benefit from unilateral deviations at the equilibrium. Computing the PBNE is challenging due to the coupling between the forward belief update and the backward policy computation. We first formulate a mathematical programming problem to compute the equilibrium policy pair under a given belief for the one-stage Bayesian game. For multi-stage Bayesian games, we compute the equilibrium policy pair under a given sequence of beliefs by constructing a sequence of nested mathematical programming problems. Finally, we combine these programs with the Bayesian update and propose an efficient algorithm to compute the PBNE.

The proposed modeling and computational methods are shown to be capable of hardening the security of a broad class of supervisory control and data acquisition (SCADA) systems. This work leverages the Tennessee Eastman process as a case study of proactive defense-in-depth strategies against the APT attackers who can infiltrate into the cyber network through phishing emails, escalate privileges through the process injection, tamper the sensor reading through malicious encrypted communication, and eventually decrease the operational efficiency of the Tennessee Eastman process without triggering the alarm. The dynamic game approach offers a quantitative way to assess the risks and provides a systematic and computational mechanism to develop proactive and strategic defenses across multiple cyber and physical stages. Based on the computation result of the case study, we obtain the following insights to guide the design of practical defense systems.

- Defense at the final stage is usually too late to be effective when APTs have been well-prepared and ready to attack. We need to take precautions and proactive responses in the cyber stages when the attack remains "under the radar" so that the attacker becomes less dominant when they reach the final stage.
- The online learning capability of the defender plays an important role in detecting the adversarial deception and tilting the information asymmetry. It increases the probability of identifying the hidden information from the observable behaviors, threatens the stealthy attacker to take more conservative actions, and hence reduces the attack loss.
- Third, defensive deception techniques are shown to be effective to introduce uncertainty to attackers, increase their learning costs, and hence reduce the probability of successful attacks. Those techniques may introduce a negative impact on

legitimate users. However, a delicate balance between security and usability can be achieved under proper designs.

1.1. Related work

One well-known industrial solution to APT defense is the ATT&CK matrix (see [Corporation, 2019](#)). It illustrates disclosed attack methods and possible detection and mitigation countermeasures at different phases of APTs. However, as argued in [Dufresne \(2018\)](#), it lacks a prioritization to list all possible attack methods in one matrix. A lot of false alarms can arise as legitimate users can also generate a majority of activities in the ATT&CK matrix. Besides, despite a persistent update, the matrix is far from complete and can lead to miss detection.

Many papers have attempted to deal with the above two challenges, i.e., false alarms and miss detection. To prevent security specialists from overwhelming alarms, [Marchetti et al. \(2016\)](#) has analyzed high volumes of network traffic to reveal weak signals of suspect APT activities and ranked these signals based on the computation of suspiciousness scores. To identify attacks that exploit zero-day vulnerabilities or other unknown attack techniques, [Friedberg et al. \(2015\)](#) has managed to learn and maintain a white-list of normal system behaviors and report all actions that are not on the white-list. There is also a rich literature on detecting essential components of an APT attack such as malicious PDF files in phishing emails (see [Nissim et al., 2015](#)), malicious SSL certificate during command and control communications (see [Ghafir et al., 2017](#)), and data leakage at the final stage of the APT campaign (see [Sigholm and Bang, 2013](#)). These works have focused on a static detection of abnormal behaviors in one specific stage but had not taken into account the correlation among multiple phases of APTs. [Ghafir et al. \(2018\)](#) has managed to build a framework to correlate alerts across multiple phases of APTs based on machine learning techniques so that all those alerts can be attributed to a single APT scenario. [Ghafir et al. \(2019\)](#) has constructed a correlation framework to link elementary alerts to the same APT campaign and applied the hidden Markov model to determine the most likely sequence of APT stages.

An alternative perspective from the aforementioned APT detection frameworks is to address how to respond to and mitigate potential attacks. [Li et al. \(2018\)](#) has captured the dynamic state evolution through a network-based epidemic model and provided both prevention and recovery strategies for defenders based on optimal control approaches. Since APTs are controlled by human experts and can act strategically, the defender's response should adapt to the potential change of APT behaviors. Thus, decision and game theory becomes a natural quantitative framework to capture constraints on defense actions, attack consequences, and attackers' incentives. [Van Dijk et al. \(2013\)](#) has proposed *FlipIt* game to model the key leakage under APTs as a private takeover between the system operator and the attacker. Many works have integrated *FlipIt* with other components for the APT defense such as the signaling game to defend cloud service (see [Pawlick et al., 2018](#)), an additional player to model the insider threats (see [Feng et al., 2015](#)), and a system of multiple nodes under limited resources (see [Zhang et al., 2015](#)). The *FlipIt* has described a high-level abstraction of the attacker's behavior to understand optimal timing for resource allocations. However, for our purpose of developing multi-stage defense policies, we need to provide a finer-grained model that can capture the dynamic interactions between players of different types across multiple stages. Our game framework models heterogeneous adversarial and defensive behaviors at multiple stages, allowing the prediction of attack moves and the estimation of losses using the equilibrium analysis.

Other security game models such as [Huang et al. \(2017\)](#); [Yang et al. \(2018\)](#); [Zhu and Rass \(2018\)](#) have provided dynamic risk

management frameworks that allow the defender to response and repair effectively. In particular, to model the multi-stage structure of APTs, [Zhu and Rass \(2018\)](#) has developed a sequence of heterogeneous game phases, i.e., a static Bayesian game for spear phishing, a nested game for penetration, and a finite zero-sum game for the final stage of physical-layer infrastructure protection. However, most of these security game frameworks have assumed complete information. Our framework explicitly models the incomplete information across the entire phases of APTs and introduces their belief updates based on multi-stage information for making long-term strategic decisions.

Cyber deception is an emerging research area. Games of incomplete information are natural frameworks to model the uncertainty and misinformation introduced by cyber deceptions. Previous works mainly focus on adversarial deceptions where the deceiver is the attacker. For example, strategic attackers in [Nguyen et al. \(2019\)](#) manipulate the attack data to mislead the defender in finitely repeated security games. A defender, on the other hand, can also initiate defensive deception techniques such as perturbations via external noises, obfuscations via revealing useless information, or honeypot deployments as shown in [Pawlick et al. \(2017\)](#). [Horák et al. \(2017\)](#) proposes a framework to engage with attackers strategically to deceive them against the attack goal without their awareness. A honeypot which appears to contain valuable information can lure attackers into isolation and surveillance. [La et al. \(2016\)](#) has used a Bayesian game to model deceptive attacks and defenses in a honeypot-enabled network in the envisioned Internet of Things. Besides detection, a honeypot can also be used to obtain high-level indicators of compromise under a proper engagement policy as shown in [Huang and Zhu \(2019a\)](#) where several security metrics are investigated and the optimal engagement policy is learned by reinforcement learning. A system can also disguise a real asset as a honeypot to evade attacks as shown in [Rowe et al. \(2007\)](#). Our work considers a dynamic Bayesian game with double-sided incomplete information to incorporate both adversarial and defensive deceptions.

The preliminary versions of this work (see [Huang and Zhu, 2018](#); [Huang and Zhu, 2019b](#)) have considered a dynamic game with one-sided incomplete information where attackers disguise themselves as legitimate users. This work extends the framework to a two-sided incomplete information structure where primitive systems can also disguise themselves as sophisticated systems. The new framework enables us to jointly investigate deceptions adopted by both attackers and defenders, and strategically design defensive deceptions to counter adversarial ones. We also develop new methodologies to address the challenge of the coupled belief update in a generalize setting without the previous assumption of the beta-binomial conjugate pair. In the case study, we investigate heterogeneous actions and cyber stages such as web phishing and privilege escalation, whose utilities are no longer negligible. Moreover, we leverage the Tennessee Eastman process with new performance metric and attack models to validate the efficacy of the proposed proactive defense-in-depth strategies, the Bayesian learning, and the defensive deception.

1.2. Organization of the paper

We summarize notations, variables, and acronyms in Table 1 for readers' convenience. We use pronoun 'he' for the user and 'she' for the defender throughout this paper. The rest of the paper is organized as follows. Section 2 introduces the multi-stage game with incomplete information and three equilibrium concepts are defined in Section 3. To compute these equilibria, we construct constrained optimization problems and an iterative algorithm in Section 4. A case study of Tennessee Eastman process under APTs

Table 1

Summary of notations, variables, and acronyms.

General notation	Meaning
$A := B$	A is defined as B
\Pr	Probability
$f: A \rightarrow B$	A function or a mapping f from domain A to codomain B
$\mathbb{E}_{a \sim A}[f(a)]$	Expectation of $f(a)$ over random variable a whose distribution is A
\mathbb{R}	Set of real numbers
$ A $	The cardinality of set A
$a \sim A$	Random variable a follows probability distribution A
$\mathbf{1}_{\{x=y\}}$	Indicator function which equals one when $x = y$, and zero otherwise
$\{a_1, \dots, a_n\}$	Set with n elements a_1, \dots, a_n
$B \setminus A$	Set of elements in B but not in A
Variable	Meaning
$i, j \in \{1, 2\}$	Index for players in the game: $i, j = 1$ for the defender and $i, j = 2$ for the user
Θ_i	Set of all possible types of player $i \in \{1, 2\}$
$\Delta(\Theta_i)$	Space of probability distributions over type set Θ_i of player $i \in \{1, 2\}$
$\theta_i \in \Theta_i$	Type of player $i \in \{1, 2\}$
θ_1^H (resp. θ_1^L)	The defender is sophisticated (resp. primitive)
θ_2^b (resp. θ_2^g)	The user is adversarial (resp. legitimate)
K	Total number of stages
$k \in \{0, 1, \dots, K\}$	Stage index
$k_0 \in \{0, 1, \dots, K\}$	Index for the initial stage
A_i^k	Set of all possible actions of player $i \in \{1, 2\}$ at stage $k \in \{0, 1, \dots, K\}$
$a_i^k \in A_i^k$	Space of probability distributions over the action set A_i^k
$a_i^k \in A_i^k$	Action of player $i \in \{1, 2\}$ at stage $k \in \{0, 1, \dots, K\}$
h^k, H^k	Action history and the set of all possible action histories at stage $k \in \{0, 1, \dots, K\}$
x^k, X^k	State and the set of all possible states at stage $k \in \{0, 1, \dots, K\}$
f^k	State transition function at stage k , i.e., $x^{k+1} = f^k(x^k, a_1^k, a_2^k)$
l_i^k, L_i^k	Available Information and set of all available information for player i at stage k
σ_i^k, Σ_i^k	Behavioral strategy and the set of all behavioral strategies for player i at stage k
$\sigma_i^k(a_i^k l_i^k)$	Probability of player i taking action a_i^k at stage k based on the available information l_i^k
$\sigma_i^{k_0:K}$	Player i 's behavioral strategies from stage k_0 to K
$\sigma_i^{*,k_0:K} (\sigma_i^{*,K} := \sigma_i^{*,K:K})$	Player i 's behavioral strategies from stage k_0 to K at the equilibrium
$b_i^k : L_i^k \mapsto \Delta(\Theta_j)$	Player i 's belief on the other player j 's type at stage k based on the available information
$b_i^k(\theta_j l_i^k)$	Probability of player j being type θ_j when player i observes information l_i^k at stage k
$J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2, w_i^k)$	Player i 's stage utility received at stage k when the state is x^k , player i takes action a_i^k , player i 's type is θ_i , and the noise is w_i^k
$J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2)$	Player i 's expected stage utility received at stage k with the input of $x^k, a_1^k, a_2^k, \theta_1, \theta_2$
$U_i^{k_0:K}(\sigma_i^{k_0:K}, \sigma_j^{k_0:K}, x^{k_0}, \theta_1, \theta_2)$	Player i 's expected cumulative utility received from stage k_0 to K when the initial state is x^{k_0} , his/her type is θ_i , and the multi-stage strategies of player i are $\sigma_i^{k_0:K}$
$V_i^k(x^k, \theta_i)$	Player i 's value function at state x^k when his/her type is θ_i
Acronym	Meaning
APT(s)	Advanced persistent threat(s)
SBNE	Static Bayesian Nash equilibrium
DBNE	Dynamic Bayesian Nash equilibrium
PBNE	Perfect Bayesian Nash equilibrium

is presented in [Section 5](#) with results in [Section 6](#). [Section 7](#) concludes the paper.

2. Dynamic game modelling of APT attacks

There are two players in the game, player 1 is the user and player 2 is the defender. The stealthy, persistent, and deceptive features of APTs result in incomplete information of the user's type to the defender. We use a finite set Θ_2 to accommodate all possible types of the user. For example, we consider a binary type set for the case study in [Sections 5](#) and [6](#) where the user's type θ_2 is either adversarial θ_2^b or legitimate θ_2^g . The APT attacker, i.e., the adversarial user, disguises himself as the legitimate user, thus the defender does not know the type of the user. The set of the user's type can also be non-binary and incorporate different APT groups when their attack tools and targeted assets are different (see [FireEye, 2017](#)).

The Defender can also be classified into different levels of sophistication based on various factors such as her level of security awareness, detection techniques she adopted, and the completeness of her virus signature database. The discrete type θ_1 distinguishes defenders of different sophistication levels and all the pos-

sible type values constitute the defender's type set Θ_1 . For example, in our case study, the defender's type θ_1 is either sophisticated θ_1^H or primitive θ_1^L . The defender can apply defensive deception techniques and keep her type private to the user. We assume that both players' type sets are commonly known. Each player knows his/her own type, yet not the other player's type. Thus, each player i should treat the other player's type as a random variable with an initial distribution b_i^0 and update the distribution to b_i^k when obtaining new information at each stage k . We present the above belief update formally in [Section 2.3](#).

2.1. Multi-stage transition

We formulate the interaction between the multi-stage APT attack and the cross-stage proactive defense into K stages of sequential games with incomplete information, as shown in [Fig. 2](#). At each stage $k \in \{0, 1, \dots, K\}$, player $i \in \{1, 2\}$ takes an action $a_i^k \in A_i^k$ from a finite and discrete set A_i^k . An intrusion detection system generates alerts based on the user's actions. However, since legitimate users can also trigger these alerts, each alert itself does not reveal the user's type. For example, an APT attacker uses the Tor network connection for data exfiltration, yet a legitimate user

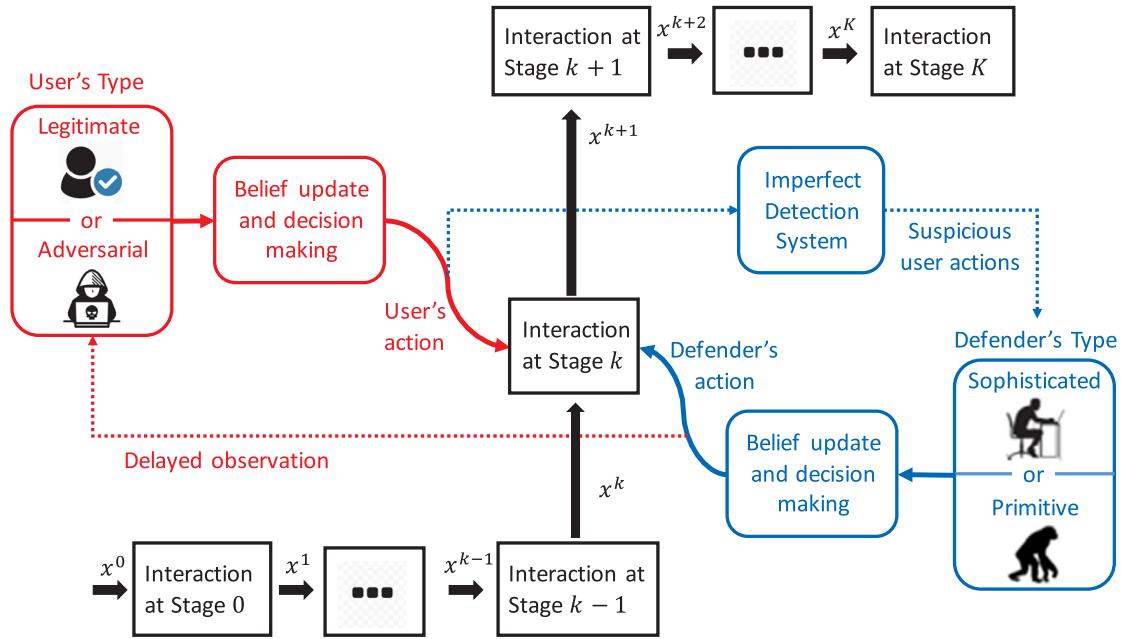


Fig. 2. A block diagram of applying the defense-in-depth approach against multi-stage APT attacks. We denote the user, the defender, and the system states in red, blue, and black, respectively. The defender interacts with the user from stage 0 to stage K in sequence where the output state of stage $k-1$ becomes the input state of stage k . At each stage k , the user observes the defender's actions at previous stages, forms a belief on the defender's type, and takes an action. At the same time, the defender makes decisions based on the output of an imperfect detection system. The dotted line means that the observation is not in real time, i.e., both players can only observe the previous-stage actions of the other player. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

can also use it legally for the traffic confidentiality as shown in Milajerdi and Kharrazi (2015). Another example is that code obfuscation can be either used legitimately to prevent reverse engineering or illegally to conceal malicious JavaScript code from being recognized by signature-based detectors or human analysts as shown in Nissim et al. (2015). We assume that the user can observe the defender's stage- k action at stage $k+1$. The observation of the defender's action at a single stage also does not reveal the defender's type.

In this paper, each player obtains a one-stage delayed observation of the other player's actions, i.e., at each stage k , the action history available to both players is $h^k = \{a_1^0, \dots, a_1^{k-1}, a_2^0, \dots, a_2^{k-1}\} \in H^k := \prod_{i=1}^2 \prod_{k=0}^{k-1} A_i^k$. Given history h^k at the current stage k , players at stage $k+1$ obtain an updated history $h^{k+1} = h^k \cup \{a_1^k, a_2^k\}$ after the observation of both players' actions at stage k . At each stage k , we further define a state $x^k \in X^k$ which summarizes information about both players' actions in previous stages so that the initial state $x^0 \in X^0$ and the history at stage k uniquely determine x^k through a known state transition function f^k , i.e., $x^{k+1} = f^k(x^k, a_1^k, a_2^k), \forall k \in \{0, 1, \dots, K-1\}$. States at different stages can have different meanings such as the reconnaissance outcome, the user's location, the privilege level, and the sensor status.

2.2. Behavioral strategy

A defender should behave differently when interacting with adversarial users and legitimate ones. The defensive measure should also vary for attackers who adopt different code families and tools. However, since the defender is uncertain about the user's type throughout the entire stages of games, she has to make judicious decisions at each stage to balance usability versus security. The user's action should also adapt to the type of the defender. For example, if the defender is primitive, an attacker prefers to take aggressive adversarial actions to achieve a quicker and low-cost compromise. However, if the defender is sophisticated and can detect

the malware with better accuracy, an attacker has to take conservative actions to remain stealthy. Since the proactive defense actions across the entire stages can affect legitimate users, they also need to be designed to avoid collateral damage.

Thus, the decision-making problem of the defender or the user boils down to the determination of a behavioral strategy $\sigma_i^k \in \Sigma_i^k : L_i^k \mapsto \Delta(A_i^k)$, i.e., player i at each stage k needs to decide which action to take or take an action with what probability based on the information $l_i^k \in L_i^k$ available to him/her at stage k . We present two different information structures in Section 2.3.1 and 2.3.2. The strategy is called 'behavioral' as the strategy depends on the information available at the time the players make their decisions. In this work, players are allowed to take *mixed strategies*, thus the co-domain of the strategy function σ_i^k is $\Delta(A_i^k)$, a probability distribution over the action space A_i^k . With a slight abuse of notation, we denote $\sigma_i^k(a_i^k | l_i^k)$ as the probability of player i taking action $a_i^k \in A_i^k$ given the available information $l_i^k \in L_i^k$. The actual action of player i taken at stage k , i.e., a_i^k , is a realization of the behavioral strategy σ_i^k . Note that the values of the other player's type θ_j and action a_j^k are not observable for player i at stage k , thus do not affect player i 's behavioral strategy σ_i^k , i.e., $\Pr(a_i^k | a_j^k, \theta_j, l_i^k) = \sigma_i^k(a_i^k | l_i^k)$. Therefore, σ_i^k and σ_j^k are conditionally independent, i.e., $\Pr(a_i^k, a_j^k | l_i^k, l_j^k) = \sigma_i^k(a_i^k | l_i^k) \sigma_j^k(a_j^k | l_j^k)$.

2.3. Belief and Bayesian update

To quantify the uncertainty of the other player's type throughout the entire stages, each player i forms a belief $b_i^k : L_i^k \mapsto \Delta(\Theta_j), j \neq i$. Likewise, $b_i^k(\theta_j | l_i^k)$ means that given information $l_i^k \in L_i^k$ at stage k , player i forms a belief that the other player j is of type $\theta_j \in \Theta_j$ with probability $b_i^k(\theta_j | l_i^k)$. At the initial stage $k=0$, the only information available to player i is his/her own type, i.e., $l_i^0 = \theta_i$. We assume that player i has a prior belief distribution b_i^0 based on the past experiences with the other player. If no pre-

vious experiences are available to player i , player i can take the uniform distribution as an unbiased prior belief. As each player i obtains new information when arriving at the next stage, his or her belief can be updated using the Bayesian rule. We present the Bayesian update under two different information structures L_i^k at stage $0 < k \leq K$ in the following two subsections.

2.3.1. Timely observations

The most straightforward information structure is $L_i^k = H^k \times \Theta_i$, i.e., the information available to player i at stage k is the action history h^k and player i 's own type θ_i , which leads to the belief update in (1).

$$\begin{aligned} b_i^{k+1}(\theta_j|h^k \cup \{a_i^k, a_j^k\}, \theta_i) \\ = \frac{\sigma_i^k(a_i^k|h^k, \theta_i)\sigma_j^k(a_j^k|h^k, \theta_j)b_i^k(\theta_j|h^k, \theta_i)}{\sum_{\bar{\theta}_j \in \Theta_j} \sigma_i^k(a_i^k|h^k, \theta_i)\sigma_j^k(a_j^k|h^k, \bar{\theta}_j)b_i^k(\bar{\theta}_j|h^k, \theta_i)}, \\ i, j \in \{1, 2\}, j \neq i. \end{aligned} \quad (1)$$

Here, player i updates the belief b_i^k based on the observation of the action a_i^k, a_j^k . When the denominator is 0, the history h^{k+1} is not reachable from h^k , and the Bayesian update does not apply. In this case, we let $b_i^{k+1}(\theta_j|h^k \cup \{a_i^k, a_j^k\}, \theta_i) := b_i^0(\theta_j|\theta_i)$.

2.3.2. Markov belief

If the information available to player i at stage k is the state value x^k and player i 's own type θ_i , then the information set is taken to be $L_i^k = X^k \times \Theta_i$. With the Markov property that $\Pr(x^{k+1}|\theta_j, x^k, \dots, x^1, x^0, \theta_i) = \Pr(x^{k+1}|\theta_j, x^k, \theta_i)$, the Bayesian update between two consequent states is

$$b_i^{k+1}(\theta_j|x^{k+1}, \theta_i) = \frac{\Pr(x^{k+1}|\theta_j, x^k, \theta_i)b_i^k(\theta_j|x^k, \theta_i)}{\sum_{\bar{\theta}_j \in \Theta_j} \Pr(x^{k+1}|\bar{\theta}_j, x^k, \theta_i)b_i^k(\bar{\theta}_j|x^k, \theta_i)}, \quad i, j \in \{1, 2\}, j \neq i. \quad (2)$$

With the conditional independence of σ_1^k and σ_2^k ,

$$\Pr(x^{k+1}|\theta_j, x^k, \theta_i) = \sum_{\{a_1^k, a_2^k\} \in \bar{A}^k} \sigma_1^k(a_1^k|x^k, \theta_1)\sigma_2^k(a_2^k|x^k, \theta_2), \quad (3)$$

where $\bar{A}^k := \{a_1^k \in A_1^k, a_2^k \in A_2^k | x^{k+1} = f^k(x^k, a_1^k, a_2^k)\}$ contains all the action pairs that change the system state from x^k to x^{k+1} . Equation (3) shows that the Bayesian update in (2) can be obtained from (1) by clustering all the action pairs in set \bar{A}^k . Thus, the Markov belief update (2) can also be regarded as an approximation of (1) using action aggregations. Unlike the history set H^k , the dimension of the state set, $|X^k|$, does not grow with the number of stages. Hence, the Markov approximation significantly reduces the memory and computational complexity. The following sections adopt the Markov belief update.

2.4. Stage and cumulative utility

The player's utility can vary under the same action taken by different types of users or defenders. For example, the remote access from a legitimate teleworker brings a reward to the defender while the one from an adversarial user inflicts a loss. Therefore, at each stage k , player i 's stage utility $J_i^k : X^k \times A_1^k \times A_2^k \times \Theta_1 \times \Theta_2 \times \mathbb{R} \mapsto \mathbb{R}$ can depend on both players' types and actions, the current state $x^k \in X^k$, and an external noise $w_i^k \in \mathbb{R}$ with a known probability density function ϖ_i^k . The noise term models unknown or uncontrollable factors that can affect the value of the stage utility. The existence of the external noise makes it impossible for player i , after reaching stage $k+1$, to infer the value of the other player's type θ_j based on the knowledge of the input parameters $x^k, a_1^k, a_2^k, \theta_i$, together with the output of the utility function J_i^k at stage k .

We denote the expected stage utility as $J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2) := \mathbb{E}_{w_i^k \sim \varpi_i^k} [J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2, w_i^k)]$, $\forall x^k, a_1^k, a_2^k, \theta_1, \theta_2$. Given the type $\theta_i \in \Theta_i$, the initial state $x^{k_0} \in X^{k_0}$, and both players' strategies $\sigma_i^{k_0:K} := [\sigma_i^k(a_i^k|x^k, \theta_i)]_{k=k_0, \dots, K} \in \prod_{k=k_0}^K \Sigma_i^k$ from stage k_0 to K , we can determine the expected cumulative utility $U_i^{k_0:K}$ for player i , i.e.,

$$\begin{aligned} U_i^{k_0:K}(\sigma_i^{k_0:K}, \sigma_j^{k_0:K}, x^{k_0}, \theta_i) \\ := \sum_{k=k_0}^K \mathbb{E}_{\theta_j \sim b_i^k, a_1^k \sim \sigma_i^k, a_2^k \sim \sigma_j^k} [J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2)] \\ = \sum_{k=k_0}^K \sum_{\theta_j \in \Theta_j} b_i^k(\theta_j|x^k, \theta_i) \sum_{a_1^k \in A_1^k} \sigma_i^k(a_i^k|x^k, \theta_i) \\ \cdot \sum_{a_2^k \in A_2^k} \sigma_j^k(a_j^k|x^k, \theta_j) J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2), \quad j \neq i. \end{aligned} \quad (4)$$

3. PBNE and dynamic programming

The user and the defender use the Bayesian update to reduce their uncertainties on the other player's type. Since their actions affect the belief update, both players at each stage should optimize their expected cumulative utilities concerning the updated beliefs at the future stages, which leads to the Perfect Bayesian Nash Equilibrium (PBNE) in Definition 1.

Definition 1. Consider the two-person K -stage game with double-sided incomplete information (i.e., each player's type is not known to the other player), a sequence of beliefs $b_i^k, \forall k \in \{0, \dots, K\}$, an expected cumulative utility $U_i^{0:K}$ in (4), and a given scalar $\varepsilon \geq 0$. A sequence of strategies $\sigma_i^{*,0:K} \in \prod_{k=0}^K \Sigma_i^k$ is called ε -dynamic Bayesian Nash equilibrium for player i if condition (C2) is satisfied. If condition (C1) is also satisfied, $\sigma_i^{*,0:K}$ is further called ε -perfect Bayesian Nash equilibrium.

- (C1) Belief consistency: Under strategy pair $(\sigma_1^{*,0:K}, \sigma_2^{*,0:K})$, each player's belief b_i^k at each stage $k = 0, \dots, K$ satisfies (2).
- (C2) Sequential rationality: For all given initial state $x^{k_0} \in X^{k_0}$ at every initial stage $k_0 \in \{0, \dots, K\}$,
$$\begin{aligned} U_1^{k_0:K}(\sigma_1^{*,k_0:K}, \sigma_2^{*,k_0:K}, x^{k_0}, \theta_1) + \varepsilon \\ \geq U_1^{k:K}(\sigma_1^{k_0:K}, \sigma_2^{*,k_0:K}, x^{k_0}, \theta_1), \quad \forall \sigma_1^{k_0:K} \in \prod_{k=0}^K \Sigma_1^k; \\ U_2^{k_0:K}(\sigma_1^{*,k_0:K}, \sigma_2^{*,k_0:K}, x^{k_0}, \theta_2) + \varepsilon \\ \geq U_2^{k:K}(\sigma_1^{*,k_0:K}, \sigma_2^{k_0:K}, x^{k_0}, \theta_2), \quad \forall \sigma_2^{k_0:K} \in \prod_{k=0}^K \Sigma_2^k. \end{aligned} \quad (5)$$

When $\varepsilon = 0$, the two ε -equilibria are called Dynamic Bayesian Nash Equilibrium (DBNE) and Perfect Bayesian Nash Equilibrium (PBNE), respectively.

The belief consistency emphasizes that when strategic players make long-term decisions, they have to consider the impact of their actions on their opponent's beliefs at future stages. The PBNE is a refinement of the DBNE with the additional requirement of the belief consistency property. When the horizon $K = 0$, the multi-stage game of incomplete information defined in Section 2 degenerates to a one-stage (static) Bayesian game with the one-stage belief pairs (b_1^K, b_2^K) and the solution concept of the DBNE/PBNE degenerates to the Static Bayesian Nash Equilibrium (SBNE) in Definition 2.

The sequential rationality property in (5) guarantees that unilateral deviations from the equilibrium at any states do not benefit the deviating player. Thus, the equilibrium strategy can be a

reasonable prediction of both players' multi-stage behaviors. DBNE strategies have the property of *strongly time consistency* because (5) holds for any possible initial states, even for states that are not on the equilibrium path, i.e., those states would not be visited under DBNE strategies. The *strongly time consistency* property makes the DBNE adapt to unexpected changes. Solutions obtained by dynamic programming naturally satisfy *strongly time consistency*. Hence, in the following, we introduce algorithms based on dynamic programming techniques.

Define the value function $V_i^{k_0}(x^{k_0}, \theta_i) := U_i^{k_0:K}(\sigma_1^{*,k_0:K}, \sigma_2^{*,k_0:K}, x^{k_0}, \theta_i)$ as the utility-to-go from any initial stage $k_0 \in \{0, \dots, K\}$ under the DBNE strategy pair $(\sigma_1^{*,k_0:K}, \sigma_2^{*,k_0:K})$. Then, at the final stage K , the value function for player $i \in \{1, 2\}$ with type θ_i at state x^K is

$$V_i^K(x^K, \theta_i) = \sup_{\sigma_i^K \in \Sigma_i^K} \mathbb{E}_{\theta_j \sim b_i^K, a_i^K \sim \sigma_i^K, a_j^K \sim \sigma_j^{*,K}} [J_i^K(x^K, a_1^K, a_2^K, \theta_1, \theta_2)]. \quad (6)$$

For any feasible sequence of belief pairs (b_1^k, b_2^k) , $k = 0, \dots, K-1$, we have the following recursive system equations for player i to find the equilibrium strategy pairs $(\sigma_1^{*,k}, \sigma_2^{*,k})$ backwardly from stage $K-1$ to the initial stage 0, i.e., $\forall k \in \{0, \dots, K-1\}$, $\forall i, j \in \{1, 2\}, j \neq i$,

$$V_i^k(x^k, \theta_i) = \sup_{\sigma_i^k \in \Sigma_i^k} \mathbb{E}_{\theta_j \sim b_i^k, a_i^k \sim \sigma_i^k, a_j^k \sim \sigma_j^{*,k}} [V_i^{k+1}(f^k(x^k, a_1^k, a_2^k), \theta_i) + J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2)]. \quad (7)$$

If we assume a virtual termination value $V_i^{K+1}(f^k(x^K, a_1^K, a_2^K), \theta_i) \equiv 0$, we can obtain (6) by letting stage $k = K$ in (7). The second term in (7) represents the immediate stage utility and the first term represents the expected utility under the future state $x^{k+1} = f^k(x^k, a_1^k, a_2^k)$, $k \in \{0, \dots, K-1\}$. Since a_i^k affects both terms, players should adopt a long-term perspective and avoid myopic behaviors to balance between the immediate utility and the expected future utility.

4. Computational algorithms

In 4.1, we formulate a constrained optimization problem to compute the SBNE and V_i^K for the one-stage game. In 4.2, we use the proposed optimization problem as building blocks to compute the DBNE and V_i^k , $\forall k \in \{0, \dots, K-1\}$. Finally, we propose an iterative algorithm to solve for the PBNE. Efficient algorithms to compute the PBNE lay a solid foundation to quantify the risk of cyber-physical attacks and guide the design of proactive defense-in-depth strategies.

4.1. One-stage Bayesian game and SBNE

Since both players' actions at the final stage $k = K$ only affect the immediate utility J_i^K and there is no future state transition, we can treat the final-stage game at each state $x^K \in X^K$ as an equivalent one-stage Bayesian game with the belief b_i^K and obtain the SBNE.

Definition 2. A pair of mixed-strategies $(\sigma_1^{*,K} \in \Sigma_1^K, \sigma_2^{*,K} \in \Sigma_2^K)$ is said to constitute a Static Bayesian Nash Equilibrium (SBNE) under the given belief pair (b_1^K, b_2^K) and the state $x^K \in X^K$, if $\forall \theta_1 \in \Theta_1, \theta_2 \in \Theta_2$,

$$\begin{aligned} & \mathbb{E}_{\theta_2 \sim b_1^K, a_1^K \sim \sigma_1^{*,K}, a_2^K \sim \sigma_2^{*,K}} [J_1^K(x^K, a_1^K, a_2^K, \theta_1, \theta_2)] \\ & \geq \mathbb{E}_{\theta_2 \sim b_1^K, a_1^K \sim \sigma_1^{*,K}, a_2^K \sim \sigma_2^{*,K}} [J_1^K(x^K, a_1^K, a_2^K, \theta_1, \theta_2)], \forall \sigma_1^K \in \Sigma_1^K; \\ & \mathbb{E}_{\theta_1 \sim b_2^K, a_1^K \sim \sigma_1^{*,K}, a_2^K \sim \sigma_2^{*,K}} [J_2^K(x^K, a_1^K, a_2^K, \theta_1, \theta_2)] \\ & \geq \mathbb{E}_{\theta_1 \sim b_2^K, a_1^K \sim \sigma_1^{*,K}, a_2^K \sim \sigma_2^{*,K}} [J_2^K(x^K, a_1^K, a_2^K, \theta_1, \theta_2)], \forall \sigma_2^K \in \Sigma_2^K. \end{aligned} \quad (8)$$

In Theorem 1, we propose a constrained optimization program C^K to compute the SBNE. We suppress the superscript of K without any ambiguity in one-stage games.

Theorem 1. A strategy pair $(\sigma_1^* \in \Sigma_1, \sigma_2^* \in \Sigma_2)$ constitutes a SBNE to the one-stage bi-matrix Bayesian game (J_1, J_2) under private type $\theta_i \in \Theta_i, \forall i \in \{1, 2\}$, belief $b_i, \forall i \in \{1, 2\}$, and a given state x , if and only if the strategy pair is a solution to C^K :

$$\begin{aligned} [C^K] : \max_{\sigma_1, \sigma_2, s_1, s_2} & \sum_{\theta_1 \in \Theta_1} \alpha_1(\theta_1) s_1(x, \theta_1) + \sum_{\theta_2 \in \Theta_2} \alpha_2(\theta_2) s_2(x, \theta_2) \\ & + \sum_{\theta_1 \in \Theta_1} \alpha_1(\theta_1) \mathbb{E}_{\theta_2 \sim b_1, a_1 \sim \sigma_1, a_2 \sim \sigma_2} [J_1(x, a_1, a_2, \theta_1, \theta_2)] \\ & + \sum_{\theta_2 \in \Theta_2} \alpha_2(\theta_2) \mathbb{E}_{\theta_1 \sim b_2, a_1 \sim \sigma_1, a_2 \sim \sigma_2} [J_2(x, a_1, a_2, \theta_1, \theta_2)] \\ \text{s.t. (a)} & \mathbb{E}_{\theta_1 \sim b_2, a_1 \sim \sigma_1} [J_2(x, a_1, a_2, \theta_1, \theta_2)] \leq -s_2(x, \theta_2), \forall \theta_2, \forall a_2, \\ & (b) \sum_{a_1 \in A_1} \sigma_1(a_1|x, \theta_1) = 1, \sigma_1(a_1|x, \theta_1) \geq 0, \forall \theta_1, \\ & (c) \mathbb{E}_{\theta_2 \sim b_1, a_2 \sim \sigma_2} [J_1(x, a_1, a_2, \theta_1, \theta_2)] \leq -s_1(x, \theta_1), \forall \theta_1, \forall a_1, \\ & (d) \sum_{a_2 \in A_2} \sigma_2(a_2|x, \theta_2) = 1, \sigma_2(a_2|x, \theta_2) \geq 0, \forall \theta_2. \end{aligned}$$

The dimensions of decision variables $\sigma_1(a_1|x, \theta_1)$, $\forall \theta_1 \in \Theta_1$, and $\sigma_2(a_2|x, \theta_2)$, $\forall \theta_2 \in \Theta_2$, are $|A_1| \times |\Theta_1|$ and $|A_2| \times |\Theta_2|$, respectively. Besides, $s_1(x, \theta_1)$, $\forall \theta_1$ and $s_2(x, \theta_2)$, $\forall \theta_2$ are scalar decision variables for each given θ_i , $i \in \{1, 2\}$. The non-decision variables $\alpha_1(\theta_1)$, $\forall \theta_1$ and $\alpha_2(\theta_2)$, $\forall \theta_2$, can be any strictly positive and finite numbers. The solution to C^K exists and is achieved at the equality of constraints (a), (c), i.e., $s_2^*(x, \theta_2) = -V_2(x, \theta_2)$, $s_1^*(x, \theta_1) = -V_1(x, \theta_1)$.

Proof. The finiteness and discreteness of the action and the type spaces guarantee the existence of the SBNE in mixed strategies as shown in Shoham and Leyton-Brown (2008), which further guarantee that program C^K has solutions. To show the equivalence between the solution to C^K and the SBNE, we first show that every SBNE is a solution of C^K . If $(\sigma_1^* \in \Sigma_1, \sigma_2^* \in \Sigma_2)$ is a SBNE pair, then the quadruple $\sigma_1^*(\theta_1), \sigma_2^*(\theta_2), s_2^*(x, \theta_2), s_1^*(x, \theta_1) = -V_2(x, \theta_2), \forall \theta_1 \in \Theta_1, \forall i \in \{1, 2\}$, is feasible because it satisfies constraints (a), (b), (c), (d). Constraints (a) and (c) imply a non-positive objective function of C^K . Since the value of the objective function achieved under this quadruple is 0, this quadruple is also optimal. Second, we show that $\sigma_1^*(\theta_1), \sigma_2^*(\theta_2), s_2^*(x, \theta_2), s_1^*(x, \theta_1)$, the result of C^K is a SBNE. The solution of C^K should satisfy all the constraints, i.e.,

$$\begin{aligned} \mathbb{E}_{\theta_1 \sim b_2, a_1 \sim \sigma_1^*, a_2 \sim \sigma_2} [J_2(x, a_1, a_2, \theta_1, \theta_2)] & \leq -s_2^*(x, \theta_2), \forall \theta_2, \forall \sigma_2 \in \Sigma_2, \\ \mathbb{E}_{\theta_2 \sim b_1, a_1 \sim \sigma_1, a_2 \sim \sigma_2^*} [J_2(x, a_1, a_2, \theta_1, \theta_2)] & \leq -s_1^*(x, \theta_1), \forall \theta_1, \forall \sigma_1 \in \Sigma_1. \end{aligned} \quad (9)$$

In particular, if we pick $\sigma_i(\theta_i) = \sigma_i^*(\theta_i), \forall \theta_i, \forall i \in \{1, 2\}$, and combine the fact that the optimal value is achieved at 0, the inequality turns out to be an equality and equation (9) becomes (8), which shows that $(\sigma_1^* \in \Sigma_1, \sigma_2^* \in \Sigma_2)$ is a SBNE. \square

Theorem 1 focuses on the double-sided Bayesian game where each player player i has a private type $\theta_i \in \Theta_i$. To accommodate the one-sided Bayesian game where player i 's type $\theta_i \in \Theta_i$ is known by both players and player j 's type remains unknown to player i , we can modify program C^K by letting $\alpha_i(\theta_i) > 0$ and $\alpha_i(\theta_i) = 0, \forall \theta_i \in \Theta_i \setminus \{\theta_i\}$.

4.2. Multi-stage Bayesian game and PBNE

From (7), we can see that at stages $k < K$, each player optimizes the sum of the immediate utility J_i^k and the utility-to-go V_i^k . Thus, we can replace the original stage utility J_i^K in program C^K with $V_i^k +$

J_i^k in program C^k to compute the DBNE in a multi-stage Bayesian game.

Theorem 2. Given a sequence of beliefs b_i^k for each player $i \in \{1, 2\}$ at each stage $k \in \{0, 1, \dots, K-1\}$, a strategy pair $(\sigma_1^{*,0:K-1}, \sigma_2^{*,0:K-1})$ constitutes a DBNE of the K -stage Bayesian game under double-sided incomplete information with the expected cumulative utility $U_i^{0:K}$ in (4), if, and only if $\sigma_1^{*,k}, \sigma_2^{*,k}, s_1^{*,k}(x^k, \theta_1), s_2^{*,k}(x^k, \theta_2)$ are the optimal solutions to the following constrained optimization problem C^k for each $k \in \{0, 1, \dots, K-1\}$:

$$\begin{aligned} [C^k] : \max_{\sigma_1^k, \sigma_2^k, s_1^k, s_2^k} & \sum_{i=1}^2 \sum_{\theta_i \in \Theta_i} \alpha_i(\theta_i) \{s_i^k(x^k, \theta_i) \\ & + \sum_{\theta_j \in \Theta_j} b_i^k(\theta_j | x^k, \theta_i) \sum_{a_1^k \in A_1^k} \sigma_1^k(a_1^k | x^k, \theta_1) \sum_{a_2^k \in A_2^k} \sigma_2^k(a_2^k | x^k, \theta_2) \\ & \cdot [J_i^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2) + V_i^{k+1}(f^k(x^k, a_1^k, a_2^k), \theta_i)]\} \\ \text{s.t. } (a) & \sum_{\theta_1 \in \Theta_1} b_2^k(\theta_1 | x^k, \theta_2) \sum_{a_1^k \in A_1^k} \sigma_1^k(a_1^k | x^k, \theta_1) \\ & \cdot [J_2^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2) + V_2^{k+1}(f^k(x^k, a_1^k, a_2^k), \theta_2)] \\ & \leq -s_2^k(x^k, \theta_2), \forall \theta_2 \in \Theta_2, \forall a_2^k \in A_2^k, \\ (b) & \sum_{\theta_2 \in \Theta_2} b_1^k(\theta_2 | x^k, \theta_1) \sum_{a_2^k \in A_2^k} \sigma_2^k(a_2^k | x^k, \theta_2) \\ & \cdot [J_1^k(x^k, a_1^k, a_2^k, \theta_1, \theta_2) + V_1^{k+1}(f^k(x^k, a_1^k, a_2^k), \theta_1)] \\ & \leq -s_1^k(x^k, \theta_1), \forall \theta_1 \in \Theta_1, \forall a_1^k \in A_1^k. \end{aligned}$$

Similarly, $\alpha_1(\theta_1), \alpha_2(\theta_2)$ can be any strictly positive and finite numbers, and $(s_1^k(x^k, \theta_1), s_2^k(x^k, \theta_2))$ is a sequence of scalar variables for each $x^k \in X^k, \theta_i \in \Theta_i, i \in \{1, 2\}$. The optimum exists and is achieved at the equality of constraints (a), (b), i.e., $s_i^{*,k}(x^k, \theta_i) = -V_i^k(x^k, \theta_i), \forall \theta_i \in \Theta_i, \forall i \in \{1, 2\}$.

The proof is similar to the one for **Theorem 1**. The decision variables σ_i^k are of size $|A_i^k| \times |X^k| \times |\Theta_i|$. By letting stage $k = K$ and $V_i^{K+1} = 0$, program C^K for the static Bayesian game is a special case of C^k for the multi-stage Bayesian game. We can solve program C^{k+1} to obtain the DBNE strategy pair $(\sigma_1^{k+1}, \sigma_2^{k+1})$ and the value of V_i^{k+1} . Then, we apply V_i^{k+1} in program C^k to obtain a DBNE strategy pair (σ_1^k, σ_2^k) and the value of V_i^k . Thus, for any given sequences of type belief pairs $b_i^k, \forall i \in \{1, 2\}, \forall k \in \{0, 1, \dots, K\}$, we can solve C^k from $k = K$ to $k = 0$ recursively to obtain the DBNE pair $(\sigma_1^{*,0:K-1}, \sigma_2^{*,0:K-1})$.

4.2.1. PBNE

Given a sequence of beliefs, we can obtain the corresponding DBNE via C^k in a backward fashion. However, given a sequence of policies, both players forwardly update their beliefs at each stage by (2). Thus, we need to find a consistent pair of belief and policy sequences as required by the PBNE. As summarized in **Algorithm 1**, we iteratively alternate between the forward belief update and the backward policy computation to find the PBNE. We resort to ε -PBNE solutions when the existence of PBNE is not guaranteed.

Algorithm 1 provides a computational approach to find ε -PBNE with the following procedure. First, both players initialize their beliefs b_i^k for every state x^k at stage $k \in \{0, 1, \dots, K\}$, according to their types. Then, they compute the DBNE strategy pair $\sigma_i^{*,0:K-1}, \forall i \in \{1, 2\}$, under the given belief sequence at each stage by solving program C^k from stage K to stage 0 in sequence. Next, they update their beliefs at each stage according to the strategy pair $\sigma_i^{*,0:K-1}, \forall i \in \{1, 2\}$, via the Bayesian update (2). If the strategy pair $\sigma_i^{*,0:K-1}, \forall i \in \{1, 2\}$, satisfies (5) under the updated belief, we find the ε -PBNE and terminate the iteration. Otherwise, we repeat the

Algorithm 1: Numerical solution of ε -PBNE.

```

1 Initialization beliefs  $b_i^k$  at each stage  $k \in \{0, 1, \dots, K\}$ ,  

  ITERNUM > 0,  $\varepsilon \geq 0$ .while the  $t <$  ITERNUM do
2    $t := t + 1$ ;
3   for each  $x^K \in X^K$  do
4     | Compute SBNE strategy  $\sigma_i^{*,K}$  and  $V_i^K(x^K, \theta_i)$  via  $C^K$ .
5   end
6   for  $k \leftarrow K - 1$  to 0 do
7     | for each  $x^k \in X^k$  do
8       | | Compute DBNE strategy  $\sigma_i^{*,k}$  and  $V_i^k(x^k, \theta_i)$  via  $C^k$ .
9     | end
10   end
11   for  $k \leftarrow 0$  to  $K - 1$  do
12     | Update  $b_i^k$  with  $\sigma_i^{*,0:K-1}$  via (2).
13   end
14   if  $\sigma_i^{*,0:K-1}, \forall i \in \{1, 2\}$ , satisfy (5) then
15     | | Terminate
16   end
17 Output  $\varepsilon$ -PBNE strategy pair  $(\sigma_1^{*,0:K-1}, \sigma_2^{*,0:K-1})$  and  

  consistent beliefs  $b_i^k, \forall k \in \{0, \dots, K\}$ .

```

backward policy computation in step two and the forward belief update in step three.

5. Case study

The model presented in **Section 2** can be applied to various APT scenarios. To illustrate the framework, this section presents a specific attack scenario where the attacker stealthily initiates infection and escalates privileges in the cyber network, aiming to launch attacks on the physical plant as shown in **Fig. 3**. Three vertical columns in the left block illustrate the state transitions across three stages: the initial compromise, the privilege escalation, and the sensor compromise of a physical system. The red squares at each column represent possible states at that stage. The right block illustrates a simplified flow chart of the Tennessee Eastman Process. We use the Tennessee Eastman process as a benchmark of industrial control systems to show that attackers can strategically compromise the SCADA system and decrease the operational efficiency of a physical plant without triggering the alarm.

In this case study, we adopt the binary type space $\Theta_2 = \{\theta_2^b, \theta_2^g\}$ and $\Theta_1 = \{\theta_1^H, \theta_1^L\}$ for the user and the defender, respectively. In particular, θ_2^b and θ_2^g denote the adversarial and legitimate user, respectively; θ_1^H and θ_1^L denote the sophisticated and primitive defender, respectively. The bi-matrices in **Tables 2–4** represent both players' expected utilities at three stages, respectively. In these matrices, the defender is the *row player* and the user is the *column player*. Each entry of the matrix corresponds to players' payoffs under their action pairs, types, and the state. In particular, the two numbers in the parenthesis before the semicolon are the payoffs of the defender and the user, respectively, under type θ_2^b , while the parenthesis in after the semicolon presents the payoff of the defender and the user, respectively, under type θ_2^g .

5.1. Initial stage: Phishing emails

We use a binary set to represent whether the reconnaissance is effectual $x^0 = 1$ or not $x^0 = 0$. Effectual reconnaissance collects essential intelligence that can better support APTs for an initial entry through phishing emails. To penalize the adversarial exploitation of the open-source intelligence (OSINT) data, the defender can create

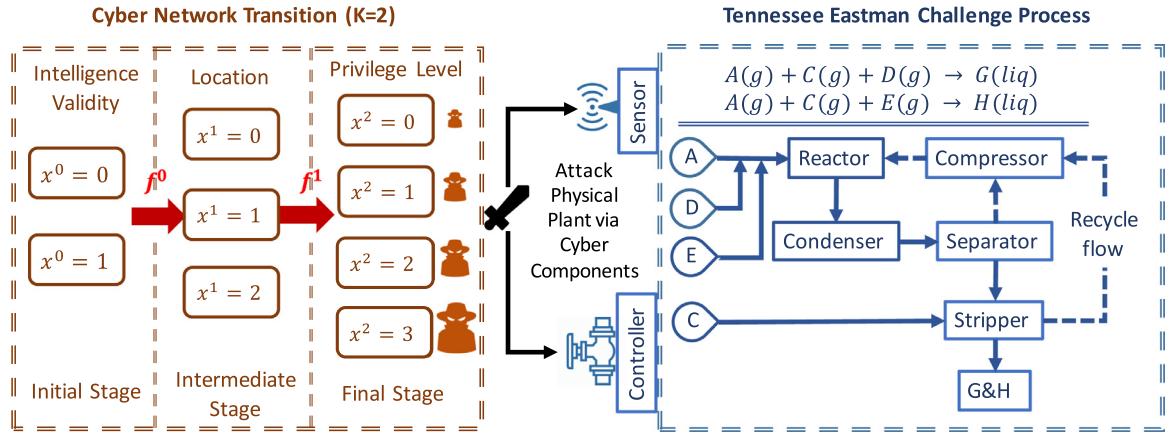


Fig. 3. The diagram of the cyber state transition (denoted by the left block in orange) and the physical attack on Tennessee Eastman process via the compromise of the SCADA system (denoted by the right block in blue). APTs can damage the normal industrial operation by falsifying controllers' setpoints, tampering sensor readings, and blocking communication channels to cause delays in either the control message or the sensing data. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
The expected utilities of the defender and the user at the initial stage, i.e., J_1^0 and J_2^0 , respectively.

θ_2^b, θ_2^g	Email employees	Email managers	Email avatars
No training	$(-r_2^0, r_2^0); (0, r_1^0)$	$(-r_2^0, r_2^0); (0, r_1^0)$	$(0, r_{b,f}^0); (0, r_{g,f}^0)$
Train employees	$(-c^0, -r^0); (-c^0, r_1^0)$	$(-c^0, r_2^0); (-c^0, r_1^0)$	$(-c^0, r_{b,f}^0); (-c^0, r_{g,f}^0)$
Train managers	$(-c^0, r_2^0); (-c^0, r_1^0)$	$(-c^0, -r^0); (-c^0, r_1^0)$	$(-c^0, r_{b,f}^0); (-c^0, r_{g,f}^0)$

Table 3

The expected utilities of the defender and the user at the intermediate stage, i.e., J_1^1 and J_2^1 , respectively.

θ_2^b, θ_2^g	NOP	Escalate privilege
Permit escalation	$(0, 0); (0, 0)$	$(-r_1^1, r_1^1); (r_1^1, r_1^1)$
Restrict escalation	$(0, 0); (0, 0)$	$(r_1^1, -r_1^1); (-r_1^1, -r_1^1)$

avatars (fake personal profiles) on the social network or the company website as shown in Molok et al. (2010).

At the initial stage of interaction, a user can send emails with non-executable attachments and shortened URLs to the accounts of entry-level employees, managers, or avatars. These three action options of the user are represented by $a_2^0 = 0, 1, 2$, respectively. Non-executable files such as PDF and MS Office are widely used in organizations yet an APT attacker can exploit them to execute malicious actions on the victim's computer. The shortened URL is created by legitimate service providers such as Google URL shortener yet can redirect to malicious links. The existing email security mechanisms are not completely effective for identifying malicious PDF files (see Nissim et al., 2015) and malicious links behind shortened URLs (see Sahoo et al., 2017). As a supplement to technical countermeasures, security training should be emphasized to increase employees' security awareness and protect them from web phishing. For example, after receiving suspicious links or attachments with strange names at unexpected times, the entry-level employee and the manager should be aware of the potential risk and apply extra security measures such as a digital signature request from the sender before clicking the link or opening the attachment. They should also be sufficiently alert and report immediately if a PDF does not contain the information that it claims to have. Then isolation can be applied to prevent the attacker from the potential lateral movement. Since employees' awareness and alertness diminish over time, the security training needs to be repeated at reasonable intervals as argued in Mitnick and Simon (2011), which can be costly. With a limited budget, the defender can choose to educate entry-level employees, manager-level

employees, or no training to avoid the prohibitive training cost c^0 . These three action options of the defender are represented by $a_1^0 = 1, 2, 0$, respectively. The utility matrix of the initial infection is given in Table 2. If the user is legitimate, i.e., $\theta_2 = \theta_2^g$, then as denoted after semicolon, he receives an immediate reward r_1^0 if he successfully communicates with the employee or the manager by email, but receives a substantial penalty $r_{g,f}^0 < 0$ if he emails the avatars because he should not contact a non-existing person. If the user is adversarial, i.e., $\theta_2 = \theta_2^b$, then as denoted before semicolon, he receives an immediate attack reward r_2^0 if the email receiver does not have proper security training, but an additional attack cost r^0 if the receiver has been trained properly. The adversarial user receives a faked reward $r_{b,f}^0 > 0$ when contacting the avatar, yet arrives at an unfavorable state at stage $k = 1$ and receives few rewards in the future stages. The training cost and the attack cost are both different for the primitive and the sophisticated defender, i.e., $c^0 := c_L^0 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^L\}} + c_H^0 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^H\}}$ and $r^0 := r_L^0 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^L\}} + r_H^0 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^H\}}$. The sophisticated defender holds the security training with a higher frequency, which incurs a higher cost, i.e., $c_H^0 > c_L^0$, but is also more effective in mitigating web phishing, i.e., $r_H^0 > r_L^0$.

5.2. Intermediate stage: Privilege escalation

The state at the intermediate stage can be interpreted as the location of the user where $x^1 = 1$ refers to the employee's computer, $x^1 = 2$ refers to the manager's computer, and $x^1 = 0$ refers to the quarantine area. After the initial access, the user operates within a process of low privilege. To access certain resources, the user needs to gain higher-level privileges. An attacker can utilize the process injection to execute malicious code in the address space of a live process and masquerade as legitimate programs to evade detection as shown in Team (2017). A mitigation method for the defender is to prevent certain endpoint behaviors that can occur during the process injection. Table 3 presents this game of privilege escalation.

The user can choose to escalate his privileges, or choose ‘no operation performed (NOP)’. The two action options are denoted by $a_2^1 = 1$ and $a_2^1 = 0$, respectively. The defender can choose to either restrict or permit an escalation, which are denoted by $a_1^1 = 1$ and $a_1^1 = 0$, respectively. If the legitimate user escalates his privilege and the defender permits escalation, then both players obtain a reward of r_1^1 . If the legitimate user escalates his privilege and the defender restricts escalation, then the efficiency reduction brings a loss of r_1^1 to both players. On the other hand, if the adversarial user escalates his privilege and the defender permits escalation, the defender receives a loss of r_2^1 . If the adversarial user escalates his privilege and the defender restricts escalation, then the adversarial user has to resort to other attack techniques which lead to a higher rate of detection. Thus, the defender obtains a reward while the attacker receives an additional cost. We assume that the reward and the additional cost are both r_L^1 if the defender is primitive, and r_H^1 if the defender is sophisticated, i.e., $r^1 = r_L^1 \cdot \mathbf{1}_{\{\theta_1=\theta_1^L\}} + r_H^1 \cdot \mathbf{1}_{\{\theta_1=\theta_1^H\}}$.

5.3. Final stage: Sensor compromise

The state at the final stage represents four possible privilege levels, denoted by $x^2 = \{0, 1, 2, 3\}$, respectively. The privilege level affects the result of the physical attack at the final stage. The defender’s and the user’s actions, and the state at the intermediate stage determine the state at the final stage. For example, if the user is at the quarantine area during the intermediate stage, then he ends up with a level-zero privilege regardless of actions taken by the defender and himself. Users who take control of the manager’s computer at the intermediate stage can obtain a higher privilege level than those who start from the entry-level employee’s computer, yet the degree of escalation is reduced if the defender chooses to restrict escalation.

We modify the Simulink model in Bathelt et al. (2015) to quantify the monetary loss of the Tennessee Eastman process under sensor compromises. Our attack model of sensor compromise is presented in Section 5.3.2. A new performance metric to quantify the operational efficiency of the Tennessee Eastman process is proposed in Section 5.3.1 and applied in the game matrix in Section 5.3.3.

5.3.1. Performance metric

The Tennessee Eastman process involves two irreversible reactions to produce two liquid (liq) products G, H from four gaseous (g) reactants A, C, D, E as shown in the right block of Fig. 3. The control objective is to maintain a desired production rate as well as quality while stabilizing the whole system under the Gaussian noise to avoid violating safety constraints such as a high reactor pressure, a high reactor temperature, and a high/low separator/stripper liquid level. Previous studies on the security of the Tennessee Eastman process have mostly focused on how an attacker can cause the shortest shutdown time (see Krofil and Cárdenas, 2013), or a serious violation of a setpoint, e.g., the reactor pressure exceeds 3,000 kpa (see Cárdenas et al., 2011). These attacks successfully cause the shutdown of the plant and a few days of shutdowns can incur a considerable financial loss. However, the shutdown also discloses the attack and leads to an immediate patch and a defense strategy update. Thus, it becomes harder for the same kind of attacks to succeed after the plant recovers from the shutdown.

In our APT scenario, the attacker aims to stealthily decrease the operational efficiency of the plant, i.e., deviate the normal operation state of the plant without triggering the safety alarm or shutting down the plant. By compromising the SCADA system and generating fraudulent sensor readings, the attacker can stealthily make the plant operates at a non-optimal state with reduced utilities.

The following economic metrics affect the operational utility of the Tennessee Eastman process:

- Hourly operating cost C_o with the unit (\$/h) is taken as the sum of purge costs, product stream costs, compressor costs, and stripper steam costs.
- Production rate R_p with the unit (m^3/h) is the volume of total products per hour.
- Quality of products Q_p with the unit (G mole%), is the percentage of G among total products.
- P_G with the unit (\$/ m^3) is the price of product G .

We propose a new performance metric U_{TE} , the *per-hour utility* to quantify the operational efficiency of the Tennessee Eastman process as follows:

$$U_{TE} = R_p \times Q_p \times P_G - C_o. \quad (10)$$

5.3.2. Attack model

An attack model is characterized by two separate parts, *information* and *capacity*. First, the information available to the attacker such as readings of different sensors can affect the performance of the attack differently. For example, observing the input rate of the raw material in the Tennessee Eastman process is less beneficial for the attacker than the direct measurements of P_G, R_p, Q_p, C_o that affect the utility metric in (10). Second, attackers can have different capacities in accessing and revising controllers and sensors. An attacker may change the parameters of the proportional-integral-derivative controller, directly falsify the controller output, or indirectly deviate the setpoint by tampering, blocking or delaying sensor readings.

In this experiment, we assume a reading manipulation of sensor XMEAS(40) and XMEAS(17) in loop 8 and loop 13 of Tennessee Eastman process (see Ricker, 1996), respectively. Sensor XMEAS(40) measures the composition of component G and sensor XMEAS(17) measures the stripper underflow. A higher privilege state $x^2 \in \{0, 1, 2, 3\}$ means that the user can access more sensors for a longer time, which results in a larger loss and thus a smaller utility of $r_1^2(x^2)$ to the defender if the user is adversarial. Fig. 4 shows the variation of U_{TE} versus the simulation time under four different privilege states. We use the time average of these utilities to obtain the normal operational utility r_4^2 and compromised utilities $r_1^2(x^2)$ under four different privilege states $x^2 \in \{0, 1, 2, 3\}$. The attacker compromises the sensor and generates fraudulent readings. The fraudulent reading can be a constant, denoted by the blue line, or a double of the real readings, denoted by the red or green lines. The pink line represents a composition attack with a limited control time. Initially, the attacker manages to compromise both sensors by doubling their readings. After the attacker loses access to XMEAS(40) at the 6th h, the system is sufficiently resilient to recover partially in about 16 h and achieve the same level of utility as the single attack in green. When the attacker also loses access to XMEAS(17) at the 36th h, the utility goes back to normal in about 13 h.

5.3.3. Utility matrix

Attacks against SCADA system can apply command injection attacks to inject false control and compromise sensor readings as shown in Morris and Gao (2013). Encryption can be introduced to conceal these malicious commands. However, a legitimate user may also encrypt his communication with the sensor to avoid eavesdropping and enhance privacy.

Therefore, at the final stage, the user has two options, sends commands to the sensor with or without encryption, which are denoted by $a_2^2 = 1$ and $a_2^2 = 0$, respectively. The defender chooses to apply either a complete or selective monitoring, denoted by

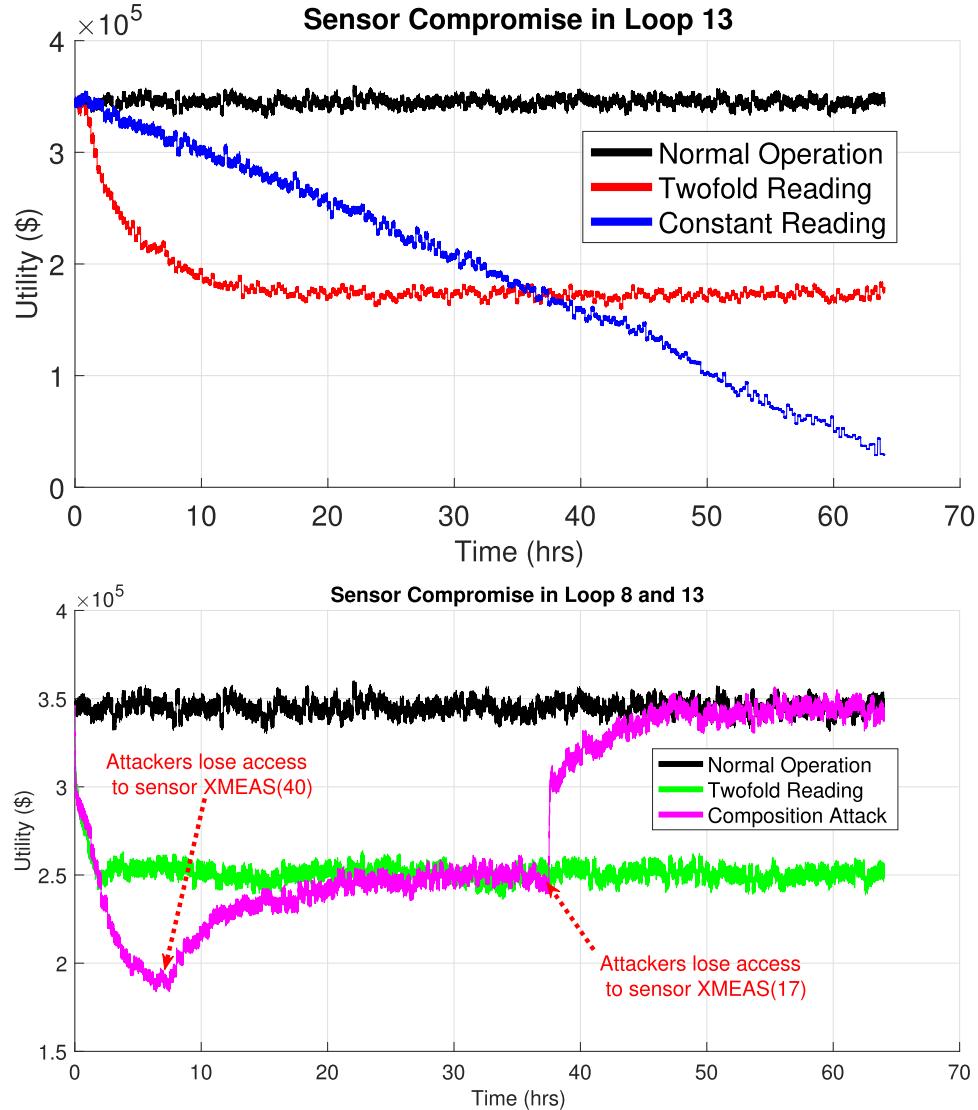


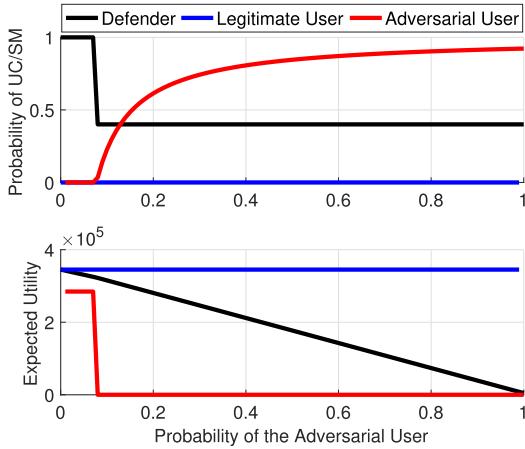
Fig. 4. The economic impact of sensor compromise in the Tennessee Eastman process. The black line represents the utility of Tennessee Eastman process under the normal operation while the other four lines represent the utility of Tennessee Eastman process under attacks with four possible privilege levels. We use the time average of these utilities to obtain the normal operational utility r_4^2 and compromised utilities $r_1^2(x^2)$, $\forall x^2 \in \{0, 1, 2, 3\}$, under four different states of privilege levels in Table 4.

Table 4
The expected utilities of the defender and the user at the final stage, i.e., J_1^2 and J_2^2 , respectively.

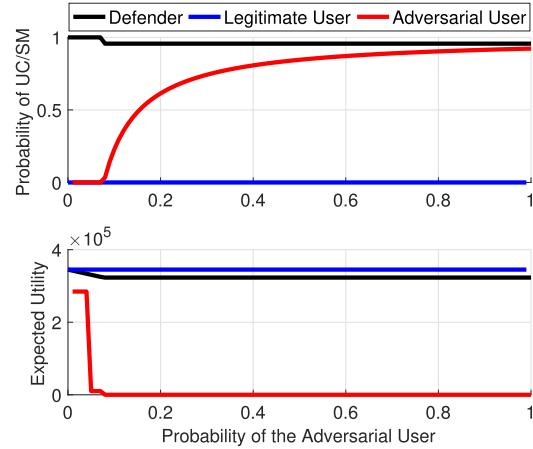
$\theta_2^b; \theta_2^g$	Unencrypted Command (UC)	Encrypted Command (EC)
Selective Monitoring (SM)	$(r_4^2, 0); (r_4^2, r_4^2/2)$	$(r_1^2(x^2), r_4^2 - r_1^2(x^2)); (r_4^2, r_4^2)$
Complete Monitoring (CM)	$(r_4^2 - c^2, 0); (r_4^2 - c^2, r_4^2/2)$	$(r_1^2 - c^2, -r^2); (r_4^2 - c^2, r_4^2)$

$a_1^2 = 1$ and $a_1^2 = 0$, respectively. The complete monitoring stores all sets of communication data and analyzes them elaborately to identify malicious commands despite encryption. The selective monitoring cannot identify malicious commands if they are encrypted. The implementation of the complete monitoring incurs an additional cost c^2 compared to the selective one. The last-stage utility matrix of both players is defined in Table 4. If the user is legitimate, as denoted in blue, both the defender and the user can receive a reward of r^4 when the Tennessee Eastman process operates normally. Legitimate users further receive a utility reduction of $r^4/2$ for the potential privacy loss if they choose unencrypted commands. For adversarial users, they send malicious commands only when the communication is encrypted to evade detection. Thus, if they choose not to encrypt the communication, they re-

ceive 0 utility and the defender receives a reward of r^4 for the normal operation. However, if they choose to send encrypted malicious commands, both players' rewards depend on whether the defender chooses the selective or complete monitoring. If the defender chooses the selective monitoring, then the adversarial user can successfully compromise the sensor, which results in a reduced utility of $r_1^2(x^2)$. In the meantime, the attacker benefits from the reward reduction of $r_4^2 - r_1^2(x^2)$. If the defender chooses the complete monitoring, then the adversarial user suffers a loss of r^2 for being detected. The detection reward and the implementation cost for two types of defenders are r_L^2, r_H^2 and c_L^2, c_H^2 , respectively. Let $r^2 := r_L^2 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^L\}} + r_H^2 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^H\}}$ and $c^2 := c_L^2 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^L\}} + c_H^2 \cdot \mathbf{1}_{\{\theta_1 = \theta_1^H\}}$.



(a) The user knows that the defender is primitive, yet the defender only knows the probability of the user being adversarial.



(b) Both players' types are private, and each player only knows the probability of the other player's type.

Fig. 5. The SBNE strategy and the expected utility of the primitive defender and the user who is either legitimate or adversarial. The y-axis of the upper figure represents the probability of either the user taking action 'selective monitoring (SM)' or the defender taking action 'unencrypted command (UC)'.

6. Computation results

In this section, we apply the algorithms introduced in Section 4 to compute both players' strategies and utilities at the equilibrium. We implement our algorithms in MATLAB and use YALMIP (see Löfberg, 2004) as the interface to call external solvers such as BARON (see Tawarmalani and Sahinidis, 2005) to solve the optimization problems. We present elaborate results from the concrete case study and provide meaningful insights of the proactive cross-layer defense against multi-stage APT attacks that are stealthy and deceptive.

For the static Bayesian game at the final stage in Section 6.1, we focus on illustrating how two players' private types affect their policies and utilities under different information structures. We further apply sensitivity analysis to show how the value of the key parameter affects the defender's and the attacker's utilities. For the multi-stage Bayesian game in 6.2, we focus on the dynamic of the belief update and state transition under the interaction of the stealthy attacker and the proactive defender. Moreover, we investigate how the adversarial and defensive deception, and how the initial state can affect the stage utility and the cumulative utility of the user and the defender.

6.1. Final stage and SBNE

Players' beliefs affect their policies and expected utilities at the final stage. We discuss three different scenarios as follows. In Fig 5a, the defender does not know the user's type. In Fig. 6, the user does not know the defender's type. In Fig. 5b, both the user and the defender do not know the other's type. In all three scenarios, the x-axis represents the belief of either the user or the defender. The y-axis of the upper figure represents the probability of either the user taking action 'selective monitoring (SM)' or the defender taking action 'unencrypted command (UC)'. Fig. 5a shows the following trends as the user becomes more likely to be adversarial. First, two black lines show that the expected utility of the defender decreases and the defender is more inclined to apply action 'complete monitoring' after her belief exceeds a threshold. Second, two red lines show that the adversarial user takes action 'unencrypted command' with a higher probability and only gains a reward when the probability of adversarial users is sufficiently small. Thus, we conclude that when the probability of the adversarial user in-

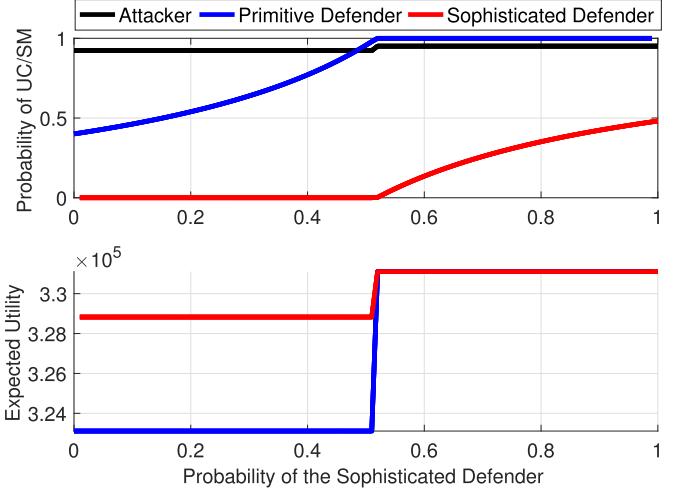


Fig. 6. The SBNE strategy and the expected utility of the adversarial user and the defender who is either primitive or sophisticated. The defender knows that the user is adversarial while the adversarial user only knows the probability of the defender being primitive. The x-axis represents the probability of the defender being sophisticated. The y-axis of the upper figure represents the probability of either the user taking action 'selective monitoring (SM)' or the defender taking action 'unencrypted command (UC)'.

creases, the defender tends to invest more in cyber defense so that the attacker behaves more conservatively and inflicts fewer losses. Third, the two blue lines show that the legitimate user always chooses 'encrypt command' and receives a constant utility, which indicates that the proactive defense does not affect the behavior and the utility of legitimate users at this stage.

Fig. 6 shows that the defender benefits from introducing defensive deception. When the defender becomes more likely to a sophisticated one, both types of defenders can have a higher probability to apply the selective monitoring and save the extra surveillance cost of the complete monitoring. The attacker with incomplete information has a threshold policy and switches to a lower attacking probability after reaching the threshold of 0.5 as shown in the black line. When the probability goes beyond the threshold, the primitive defender can pretend to be a sophisticated one and take action 'selective monitoring'. Meanwhile, a sophisticated

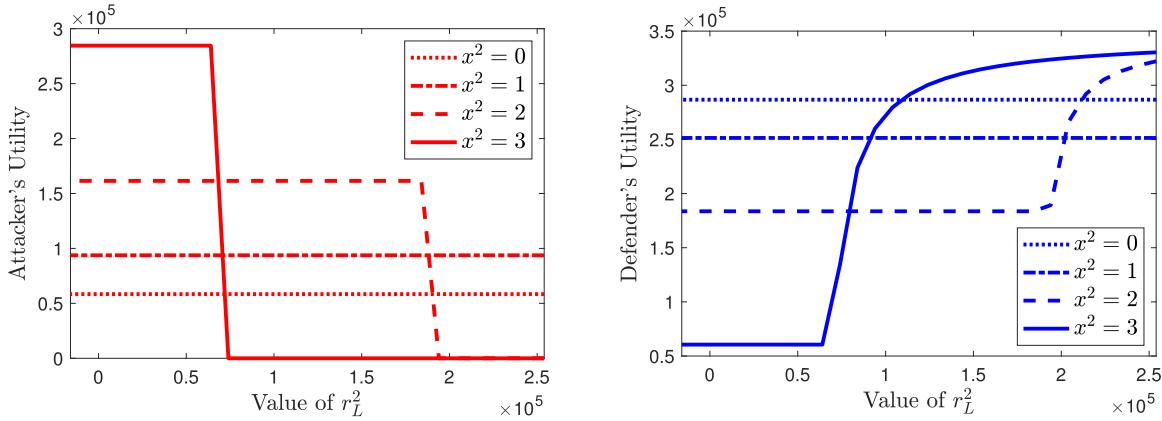


Fig. 7. Utilities of the primitive defender and the attacker versus the value of r_L^2 under different states $x^2 \in \{0, 1, 2, 3\}$.

defender can reduce the security effort and take action ‘selective monitoring’ with a higher probability since the attacker becomes more cautious in taking adversarial actions after identifying the defender as more likely to be sophisticated. It is also observed that the sophisticated defender receives a higher payoff before the attacker’s belief reaches the 0.5 threshold. After the belief reaches the threshold, the attacker is threatened to take less aggressive actions, and both types of defenders share the same payoff.

Finally, we consider the double-sided incomplete information where both players’ types are private information, and each player only has the belief of the other player’s type. Compared with the defender in Fig. 5a who takes action ‘selective monitoring’ with a probability less than 0.5 and receives a decreasing expected payoff, the defender in Fig. 5b can take ‘selective monitoring’ with a probability closed to 1 and receive a constant payoff in expectation after the user’s belief exceeds the threshold. Thus, the defender can spare defense efforts and mitigate risks by introducing uncertainties on her type as a countermeasure to the adversarial deception.

6.1.1. Sensitivity analysis

As shown in Fig. 7, if the value of the penalty r_L^2 is close to 0, i.e., the defense at the final stage is ineffective, then an arrival at state $x^2 = 3$, the highest privilege level can significantly increase the attacker’s payoff and cause the most damage to the defender. As more effective defensive methods are employed at the final stage, i.e., the value of r_L^2 increases, the attacker becomes more conservative and strategic in taking adversarial behaviors. Then, the state with the highest privilege level may not be the most favorable state for the attacker.

6.2. Multi-stage and PBNE

We show in Fig. 8 that the Bayesian belief update leads to a more accurate estimate of users’ types. Without the belief update, the posterior belief is the same as the prior belief in red and is used as the baseline. As the prior belief increases in the x-axis, the posterior belief after the Bayesian update also increases in blue. The blue line is in general above the red line, which means that with the Bayesian update, the defender’s belief becomes closer to the right type. Also, we find that the belief update is the most effective when an inaccurate prior belief is used as it corrects the erroneous belief significantly.

In Fig. 9, we show that the proactive defense, i.e., defensive methods in intermediate stages can affect the state transition and reduce the probability of attackers reaching states that can result in huge damage at the final stage. As the prior belief of the user being adversarial increases, the attacker is more likely to arrive

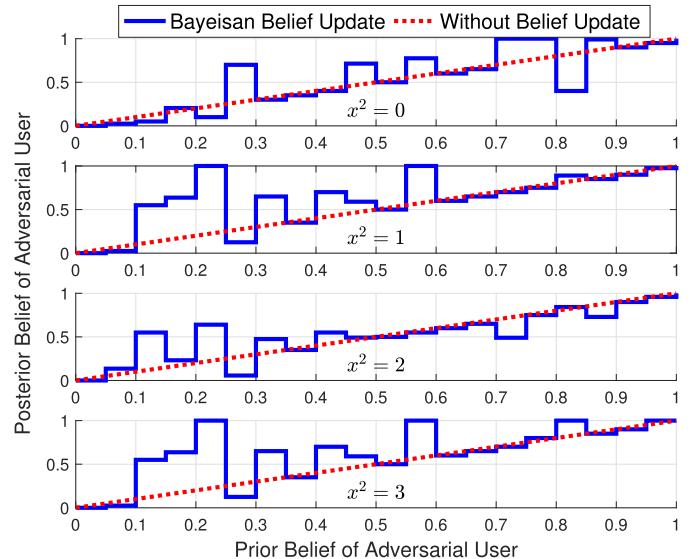


Fig. 8. The defender’s prior and posterior beliefs of the user being adversarial.

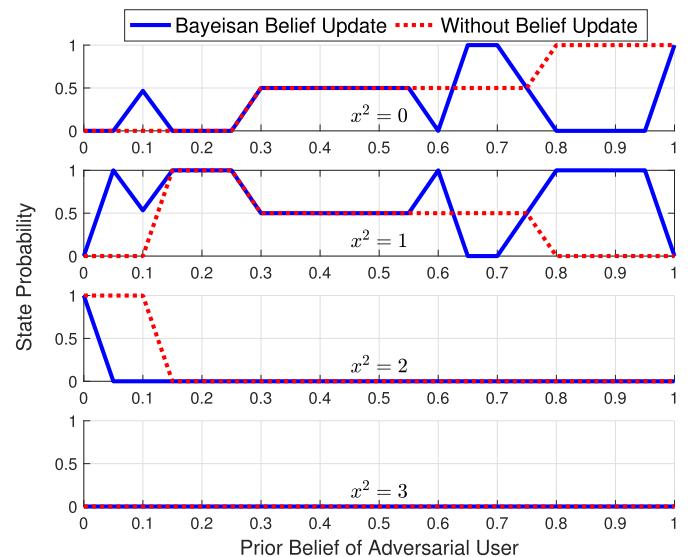


Fig. 9. The probability of different states $x^2 \in \{0, 1, 2, 3\}$.

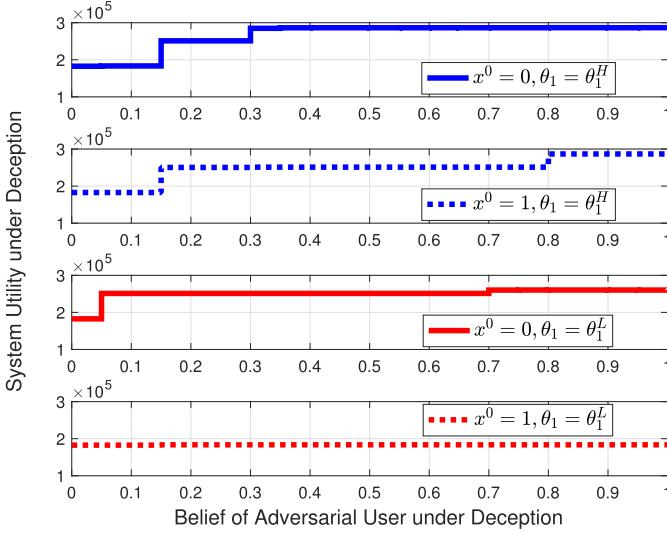


Fig. 10. The defender's utility under deceived beliefs.

at state $x^2 = 0$ and $x^2 = 1$, and reduce the probability of visiting $x^2 = 2$ and $x^2 = 3$.

6.2.1. Adversarial and defensive deception

Fig. 10 investigates the adversarial deception where the attacker takes full control of the defense system and manipulates the defender's belief. As shown in the figure, the defender's utilities all increase when the belief under the deception approaches the correct belief that the user is adversarial. Also, the increase is stairwise, i.e., the defender only alternates her policy when the ma-

nipulated belief is beyond certain thresholds. Under the same manipulated belief, a sophisticated defender benefits no less than a primitive one. The defender receives a lower payoff when the reconnaissance provides effectual intelligence.

Incapable of revealing the adversarial deception completely, the defender can alternatively introduce defensive deceptions, e.g., a primitive defender can disguise himself as a sophisticated one to confuse the attacker. Defensive deceptions introduce uncertainties to attackers, increase their costs, and increase the defender's utility.

Fig. 11 investigates the defender's and the attacker's utilities under three different scenarios. The complete information refers to the scenario where both players know the other player's type. The deception with the *H*-type or the *L*-type means that the attacker knows the defender's type to be sophisticated or primitive, respectively, yet the defender has no information about the user's type. The double-sided deception indicates that both players do not know the other player's type. The results from Fig. 11 are summarized as follows. First, the sophisticated defender's payoffs can increase as much as 56% than those of the primitive defender. Also, a prevention of effectual reconnaissance increases the defender's utility by as much as 41% and reduces the attacker's utility by as much as 38%. Second, the defender and the attacker receive the highest and the lowest payoff, respectively, under the complete information. When the attacker introduces deceptions over his type, the attacker's utility increases and the defender's utility decreases. Third, when the defender adopts defensive deceptions to introduce double-sided incomplete information, we find that the decrease of the sophisticated defender's utilities is reduced by at most 64%, i.e., changes from \$55,570 to \$35,570 when the reconnaissance is effectual. The double-sided incomplete information also brings lower utilities to the attacker than the one-sided adversarial deception.

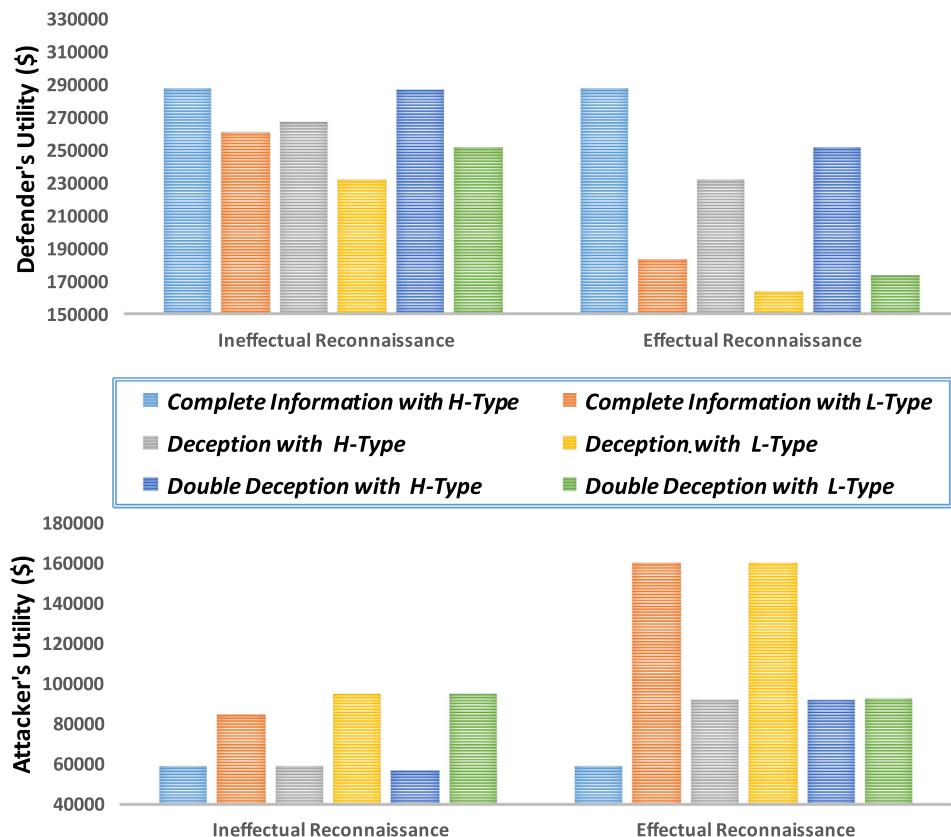


Fig. 11. The cumulative utilities of the attacker and the defender under the complete information, the adversarial deception, and the defensive deception. In the legend, the left three represent the utilities for a sophisticated defender and the right three represent the ones for a primitive defender.

However, the defender's utility under the double-sided deception is still less than the complete information case, which concludes that acquiring complete information of the adversarial user is the most effective defense. However, if the complete information cannot be obtained, the defender can mitigate her loss by introducing defensive deceptions.

7. Discussions and conclusions

Advanced Persistent Threats (APTs) are emerging security challenges for cyber-physical systems as the attacker can stealthily enter, persistently stay in, and strategically interact with the system. In this work, we have developed a game-theoretic framework to design proactive and cross-layer defenses for cyber-physical systems in a holistic manner. Dynamic games of incomplete information have been used to capture the long-term interaction between users and defenders who have private information unknown to the other player. Each player forms a belief on the unknowns and uses the Bayesian update to learn the private information and reduce uncertainty. The analysis of the Perfect Bayesian Nash Equilibrium (PBNE) has provided the defender with an effective countermeasure against the stealthy strategic attacks at multiple stages. To compute the PBNE of the dynamic games, we have proposed a nested algorithm that iteratively alternates between the forward belief update and the backward policy computation. The algorithm has been shown to quickly converge to the ε -PBNE that yields a consistent pair of beliefs and policies.

Using the Tennessee Eastman process as a case study of industrial control systems, we have shown that the proactive multi-stage defense in cyber networks can successfully mitigate the risk of physical attacks without reducing the payoffs of legitimate users. In particular, experiment results show that a sophisticated defender receives a payoff up to 56% higher than a primitive defender does. Also, it has been illustrated that by preventing effectual reconnaissance, the defender increases her utility and reduces the attacker's utility by at most 41% and 38%, respectively. On one hand, the attacker receives a higher payoff after introducing the adversarial deception as it increases the defender's uncertainties on the user's type. On the other hand, by creating uncertainties for attackers, the defender can successfully threaten them to take more conservative behaviors and become less motivated to launch attacks. It has been shown that the defender can significantly benefit from the mitigation of attack losses when he adopts defensive deceptions.

The main challenge of our approach is to identify the utility and feasible actions of defenders and users at each stage. One future direction to reduce the complexity of the model description is to develop mechanisms that can automate the synthesis of verifiably correct game-theoretic models. It would alleviate the workload of the system defender and operator. Nevertheless, game theory provides a quantitative and explainable framework to design the proactive defensive response under uncertainty compared to rule-based and machine-learning-based defense methods, respectively. Besides, the rule-based defense is static, thus an attack can circumvent it through sufficient effort. Machine learning methods require a lot of labeled data sets which may be hard to obtain in the APT scenario. Second, we have proposed the belief to quantify the uncertainty which results from players' private types. The belief is continuously updated to reduce uncertainties and provide a probabilistic detection system as a byproduct of the APT response design. Third, our approach enables the defender to evaluate the multi-stage impact of her defense strategies to both legitimate and adversarial users when adversarial and defensive deceptions present at the same time. Based on the evaluation, defenders can further find revised countermeasures and design new game rules to achieve a better tradeoff between security and usability. Our

model can be broadly applied to scenarios in artificial intelligence, economy, and social science where multi-stage interactions occur between multiple agents with incomplete information. Multi-sided non-binary types can be defined based on the scenario, and our iteration algorithm of the forward belief update and the backward policy computation can be extended for efficient computations of the perfect Bayesian Nash equilibrium. The future work would extend the framework to an N -person game to characterize the simultaneous interactions among multiple users and model composition attacks. We would also consider scenarios where players' actions and the system state are partially observable.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgements

This research is partially supported by award 2015-ST-061-CIRCO1 from U.S. Department of Homeland Security; awards ECCS-1847056, CNS-1544782, and SES-1541164 from [National Science Foundation \(NSF\)](#); and grant [W911NF-19-1-0041](#) from Army Research Office (ARO).

References

- Bathelt, A., Ricker, N.L., Jelali, M., 2015. Revision of the tennessee eastman process model. In: 9th IFAC Symposium on Advanced Control of Chemical Processes AD-CHEM 2015, 48, pp. 309–314. doi:[10.1016/j.ifacol.2015.08.199](https://doi.org/10.1016/j.ifacol.2015.08.199).
- Cárdenas, A.A., Amin, S., Lin, Z.-S., Huang, Y.-L., Huang, C.-Y., Sastry, S., 2011. Attacks against process control systems: risk assessment, detection, and response. In: Proceedings of the 6th ACM symposium on information, computer and communications security. ACM, pp. 355–366.
- Corporation, T.M., 2019. Enterprise Matrix. <https://attack.mitre.org/matrices/enterprise/>.
- Dufresne, M., 2018. Putting the MITRE ATT&CK evaluation into context. <https://www.endgame.com/blog/technical-blog/>.
- Feng, X., Zheng, Z., Hu, P., Cansever, D., Mohapatra, P., 2015. Stealthy attacks meets insider threats: a three-player game model. In: MILCOM 2015–2015 IEEE Military Communications Conference. IEEE, pp. 25–30.
- FireEye, 2017. Advanced Persistent Threat Groups | FireEye. <https://www.fireeye.com/current-threats/apt-groups.html>.
- Friedberg, I., Skopik, F., Settanni, G., Fiedler, R., 2015. Combating advanced persistent threats: from network event correlation to incident detection. Comput. Secur. 48, 35–57.
- Ghafir, I., Hammoudeh, M., Prenosil, V., Han, L., Hegarty, R., Rabie, K., Aparicio-Navarro, F.J., 2018. Detection of advanced persistent threat using machine-learning correlation analysis. Future Gener. Comput. Syst. 89, 349–359.
- Ghafir, I., Kyriakopoulos, K.G., Lambotharan, S., Aparicio-Navarro, F.J., AsSadhan, B., BinSalleh, H., Diab, D.M., 2019. Hidden Markov models and alert correlations for the prediction of advanced persistent threats. IEEE Access 7, 99508–99520.
- Ghafir, I., Prenosil, V., Hammoudeh, M., Han, L., Raza, U., 2017. Malicious ssl certificate detection: a step towards advanced persistent threat defence. In: Proceedings of the International Conference on Future Networks and Distributed Systems. ACM, p. 27.
- Harsanyi, J.C., 1967. Games with incomplete information played by "Bayesian" players, i–iii part i. the basic model. Manage. Sci. 14 (3), 159–182.
- Department of Homeland Security, D., 2018. NSA/CSS Technical Cyber Threat Framework v2 A Report from: Cybersecurity Operations the Cybersecurity Products and Sharing Division. Technical Report. <https://www.nsa.gov/Portals/70/documents/what-we-do/cybersecurity/professional-resources/crt-nsa-css-technical-cyber-threat-framework.pdf>.
- Horák, K., Zhu, Q., Bošanský, B., 2017. Manipulating adversaries belief: a dynamic game approach to deception by design for proactive network security. In: International Conference on Decision and Game Theory for Security. Springer, pp. 273–294.
- Huang, L., Chen, J., Zhu, Q., 2017. A large-scale markov game approach to dynamic protection of interdependent infrastructure networks. In: International Conference on Decision and Game Theory for Security. Springer, pp. 357–376.
- Huang, L., Zhu, Q., 2018. Analysis and computation of adaptive defense strategies against advanced persistent threats for cyber-physical systems. In: International Conference on Decision and Game Theory for Security. Springer, pp. 205–226.
- Huang, L., Zhu, Q., 2019. Adaptive honeypot engagement through reinforcement learning of semi-markov decision processes. In: International Conference on Decision and Game Theory for Security. Springer, pp. 196–216.

- Huang, L., Zhu, Q., 2019. Adaptive strategic cyber defense for advanced persistent threats in critical infrastructure networks. *ACM SIGMETRICS Performance Evaluation Review* 46 (2), 52–56.
- Hutchins, E.M., Cloppert, M.J., Amin, R.M., 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Lead. Issues Inf. WarfareSecur.Res.* 1 (1), 80.
- Krotofil, M., Cárdenas, A.A., 2013. Resilience of process control systems to cyber-physical attacks. In: *Nordic Conference on Secure IT Systems*. Springer, pp. 166–182.
- La, Q.D., Quek, T.Q., Lee, J., Jin, S., Zhu, H., 2016. Deceptive attack and defense game in honeypot-enabled networks for the internet of things. *IEEE Internet of Things J.* 3 (6), 1025–1035.
- Li, P., Yang, X., Xiong, Q., Wen, J., Tang, Y.Y., 2018. Defending against the advanced persistent threat: an optimal control approach. *Secur. Commun. Netw.* 2018.
- LLC, P. I., 2018. 2018 cost of data breach study.
- Löfberg, J., 2004. Yalmip: a toolbox for modeling and optimization in MATLAB. In: *Proceedings of the CACSD Conference*. Taipei, Taiwan.
- Marchetti, M., Pierazzi, F., Colajanni, M., Guido, A., 2016. Analysis of high volumes of network traffic for advanced persistent threat detection. *Comput. Netw.* 109, 127–141.
- Messaoud, B.I., Guennoun, K., Wahbi, M., Sadik, M., 2016. Advanced Persistent Threat: new analysis driven by life cycle phases and their challenges. In: *2016 International Conference on Advanced Communication Systems and Information Security (ACOSIS)*. IEEE, pp. 1–6.
- Milajerdi, S.M., Kharrazi, M., 2015. A composite-metric based path selection technique for the tor anonymity network. *J. Syst. Softw.* 103, 53–61.
- Mitnick, K.D., Simon, W.L., 2011. *The Art of Deception: Controlling the Human Element of Security*. John Wiley & Sons.
- Molok, N. N. A., Chang, S., Ahmad, A., 2010. Information leakage through online social networking: Opening the doorway for advanced persistence threats. In: *Proceedings of the 8th Australian Information Security Management Conference*.
- Morris, T.H., Gao, W., 2013. Industrial control system cyber attacks. In: *Proceedings of the 1st International Symposium on ICS & SCADA Cyber Security Research*, pp. 22–29.
- Nguyen, T. H., Wang, Y., Sinha, A., Wellman, M. P., Deception in finitely repeated security games, In: *33th AAAI Conference on Artificial Intelligence*, 2019.
- Nissim, N., Cohen, A., Glezer, C., Elovici, Y., 2015. Detection of malicious pdf files and directions for enhancements: a state-of-the art survey. *Comput. Secur.* 48, 246–266.
- Pawlak, J., Chen, J., Zhu, Q., 2018. ISTRCT: an interdependent strategic trust mechanism for the cloud-enabled internet of controlled things. *arXiv preprint arXiv:1805.00403*.
- Pawlak, J., Colbert, E., Zhu, Q., 2017. A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. *arXiv preprint arXiv:1712.05441*.
- Ricker, N.L., 1996. Decentralized control of the tennessee eastman challenge process. *J. Process Control* 6 (4), 205–221.
- Rowe, N. C., Custy, E. J., Duong, B. T., 2007. Defending cyberspace with fake honeypots. *JCP* 2.2 (2007), 25–36.
- Sahoo, D., Liu, C., Hoi, S. C., 2017. Malicious url detection using machine learning: a survey. *arXiv preprint arXiv:1701.07179*.
- Shoham, Y., Leyton-Brown, K., 2008. *Multiagent Systems: Algorithmic, Game-theoretic, and Logical Foundations*. Cambridge University Press.
- Sigholm, J., Bang, M., 2013. Towards offensive cyber counterintelligence: adopting a target-centric view on advanced persistent threats. In: *2013 European Intelligence and Security Informatics Conference*. IEEE, pp. 166–171.
- Tawarmalani, M., Sahinidis, N.V., 2005. A polyhedral branch-and-cut approach to global optimization. *Math. Program.* 103, 225–249.
- Team, M. D. A. R., 2017. Detecting stealthier cross-process injection techniques with windows defender atp: process hollowing and atom bombing. <https://bitly/2nVWDQd>.
- Van Dijk, M., Juels, A., Oprea, A., Rivest, R.L., 2013. Flipit: the game of “stealthy takeover”. *J. Cryptol.* 26 (4), 655–713.
- Yang, L.-X., Li, P., Zhang, Y., Yang, X., Xiang, Y., Zhou, W., 2018. Effective repair strategy against advanced persistent threat: a differential game approach. *IEEE Trans. Inf. ForensicsSecur.* 14 (7), 1713–1728.
- Zhang, M., Zheng, Z., Shroff, N.B., 2015. A game theoretic model for defending against stealthy attacks with limited resources. In: *International Conference on Decision and Game Theory for Security*. Springer, pp. 93–112.
- Zhu, Q., Rass, S., 2018. On multi-phase and multi-stage game-theoretic modeling of advanced persistent threats. *IEEE Access* 6, 13958–13971.

Linan Huang received the B.Eng. degree in Electrical Engineering from Beijing Institute of Technology, China, in 2016. He is currently a Ph.D. candidate in the Laboratory for Agile and Resilient Complex Systems, Tandon School of Engineering, New York University. His research interests include dynamic decision making of the multi-agent system, security and resilience for the cyber-physical systems.

Quanyan Zhu (S'04-M'12) received B. Eng. in Honors Electrical Engineering with distinction from McGill University in 2006, M.A.Sc. from University of Toronto in 2008, and Ph.D. from the University of Illinois at Urbana-Champaign (UIUC) in 2013. After stints at Princeton University, he is currently an assistant professor at the Department of Electrical and Computer Engineering, New York University. He is a recipient of many awards including NSF CAREER Award, NYU Goddard Junior Faculty Fellowship, NSERC Postdoctoral Fellowship (PDF), NSERC Canada Graduate Scholarship (CGS), and Mavis Future Faculty Fellowships. He spearheaded and chaired INFOCOM Workshop on Communications and Control on Smart Energy Systems (CCSES), and Midwest Workshop on Control and Game Theory (WCGT). His current research interests include resilient and secure interdependent critical infrastructures, Internet of Things, cyber-physical systems, game theory, machine learning, network optimization and control. He has served as the general chair of the 7th Conference on Decision and Game Theory for Security (GameSec) in 2016, the 9th International Conference on NETwork Games, COntrol and OPTimisation (NETCOOP) in 2018, and the 5th International Conference on Artificial Intelligence and Security (ICAIS 2019) in 2019.