A Unifying Approach for the Identification of Application-driven Stealthy Attacks on Mobile CPS

Vu Nguyen
Department of Computer Science
Texas State University
vn1035@txstate.edu

Mina Guirguis
Department of Computer Science
Texas State University
msg@txstate.edu

George Atia
Department of Electrical Engineering and Computer Science
University of Central Florida
george.atia@ucf.edu

Abstract— **Cyber-Physical Systems (CPS) employing mobile nodes rely on wireless communication in many critical applications. Due to interference and intentional jamming by adversaries, mobile nodes may fail to communicate with each other causing severe performance consequences. In this paper, we present a unifying approach for identifying attacks that target Mobile CPS applications. The attack policies are obtained as solutions to Markov Decision Process (MDP) problems, in which a decision to interfere with a signal on a given link is based on the current state of the system. Through applying approximate policy iteration methods, efficient attack policies that only interfere with a selective set of signals between the mobile nodes are derived to maximize damage while minimizing exposure and detection. The proposed approach is instantiated on pheromone-based coordination methods that are used in reconnaissance, surveillance, and search missions in military operations. The identified attack policies are shown to be more potent than other attack policies, including myopic, heuristic, and Denial of Service (DoS) policies.**

I. Introduction

Motivation: Advances in wireless communication technologies have enabled the development of various Cyber-Physical Systems (CPS) that employ mobile nodes. For example, in Intelligent Transportation Systems, communication between vehicles enables sharing critical information about the road (e.g., congestion, accidents, etc...) causing vehicles to make better decisions. In systems that employ autonomous vehicles, communication is required to coordinate tasks such as tracking targets, covering a region, and carrying out search and rescue operations. In the above systems, nodes rely on wireless channels to make decisions (whether directly through control/measurement signals received, or indirectly through a human operator). Those decisions, in return, change the overall state of the system.

Due to the shared nature of the wireless channels used, the evolution of the overall state of a Mobile CPS is affected by interference and intentional jamming by adversaries. For example, if some nodes fail to communicate, their decisions would impact the whole system in a manner that may not have been intended. Attacks on Intelligent Transportation Systems were considered in [1] where it was shown that failure of communication between vehicles and the infrastructure can lead to traffic congestion. There has also been a lot of research work that studied the impact of noise on the stability/safety/usability of the system [2]–[9]. Similarly, the effect of jamming has been shown to cause severe effects that may cripple the whole system [10]–[14]. Luckily, by their very nature, these attacks can be easily detected, allowing some countermeasures to be taken [15]–[17].

But, what if an adversary can force a Mobile CPS to evolve into inefficient states without continuous jamming? What if one of the goals of the attacks is to go undetected? And what if the adversary can optimize the attack policy over when and which links to attack?

Scope: Unlike traditional attacks that target common networking protocols used by Mobile CPS (e.g., TCP, 802.11), this paper presents a unifying approach to identify dynamic application-driven attacks. We argue that adversaries are typically interested in attacking *the application itself* rather than the protocols used by the application. By doing so, they can selectively choose the attack pattern based on their observation of the application, rather than constantly exploiting a vulnerability of a given protocol. This way not only they remain stealthy, but can rather still mount attacks even if the mobile CPS switches between protocols.

As a particular instantiation of our framework, we expose vulnerabilities in pheromone-based coordination methods used in Mobile ad-hoc CPS. Over the past decade, there has been a constant increase in the usage of drones and Unmanned Combat Vehicles (UCVs), specially in military operations [18], [19]. Pheromone-based coordination methods enable mobile nodes to act independently and yet collectively solve a given problem. Through exchanging messages depicting different pheromone levels at different locations, nodes can make cooperative decisions to visit areas more often, or provide a balanced coverage for a given region, for example. Pheromone-based coordination methods are known for their robustness against a wide range of attacks since a probabilistic approach is adopted for path planning. It is not clear, however, how would the overall mission objective be adversely affected by an attacker who can selectively attack a subset of links between the mobile nodes.

Contributions: Mobile CPS rely on wireless links that carry important control and measurement signals. Thus, it becomes important to study their susceptibility to attacks through a framework that can identify attacks – specially stealthy ones – in a systematic manner. In particular, this paper makes the following contributions:

- Unlike protocol-driven attacks, we consider dynamic application-driven attacks and provide a unifying frame-

work for the identification of optimal/suboptimal attack policies that are solutions to Markov Decision Problems (MDPs). The policies obtained capture the best interest of the attacker – minimizing the cost of the attack while maximizing the incurred damage. The attack decisions are based on the current state of the mobile nodes.
- As a case study, we investigate the susceptibility of pheromone-based coordination methods to attacks and identify different classes of stealthy attacks based on an attack cost metric. When the attack cost is very low, our exposed attacks resemble DoS attacks in which all the links are attacked, but as we increase the cost of the attacks, we identify more stealthy ones.
- We develop new metrics that capture discrepancies between the information available to the different mobile nodes, which are important from a security standpoint, since they can be used in mounting potent attacks.
- Through reliance on a set of features – that we were able to carefully craft – we were able to provide effective approximation methods for solving a high-dimensional problem that is otherwise computationally prohibitive.

Paper Organization: In Section II, we survey the related work this paper pertains to. In Section III we present the general framework used to identify attack policies on Mobile CPS. We instantiate this framework in Section IV for exposing attacks on pheromone-based coordination algorithms. We evaluate the impact of the attacks in Section V and we conclude the paper in Section VI.

II. Related Work

The research work in this paper is at the intersection of two research areas: (1) Controlling Mobile CPS and (2) Security issues raised by the loss and delay of signals between nodes.

The work in [2], [6], [20] consider general frameworks to study the impact of loss (and delay) of control and measurement signals on the links between connected components and their impact on the efficiency, stability and usability of the system. Different studies vary in their assumptions about the process involved in dropping and/or delaying signals. The authors in [3] consider the problem of optimal estimation under a Bernoulli packet dropping process through a time-varying Kalman filter. They show the existence of a particular drop rate beyond which the estimation error covariance becomes unbounded. In [4], the authors consider a packet loss process that follows a Markov model (rather than the traditional Bernoulli model). They develop a predictor in which the gain is selected online to cope with the losses. In [5], the impact of deterministic dropping rates on the performance of an optimal controller is assessed. Therein sufficient conditions to ensure the stability of the system subject to certain packet loss rates and delays are derived. The problem of control and estimation under the effect of common networking protocols (e.g., TCP and UDP) is studied in [6]. Different studies advocate different actions for missing measurement and control packets. In [7], the authors consider a controller that would use zero values for missed measurement packets. Alternatively, in [8], the use of timers to generate new control signals in the absence of fresh measurement packets is considered. The new control signals are predicted based on the last value(s) of the applied ones.

Recently there has been a pressing growing concern for securing control systems [21], [22] with many research challenges to be tackled [23], [24]. The most related to our work is the research work in [25], [26]. The authors in [25] study the performance of a linear control system subject to various Denial of Service (DoS) attack models on the measurements and control signals (e.g., random Bernoulli, constrained and general). They investigate the design of an optimal feedback controller that minimizes the damage from various DoS attacks. The study, however, was limited to a single linear feedback loop and without considering a specific application to assess the potential impacts. In [26], the authors consider the effect of surge, bias and geometric attacks on a linear feedback control system for a Tennessee Eastman process control system. The study, however, was limited to a specific set of attacks without aiming to explore the whole attack space. Other work investigated the impact of cyber attacks on control systems although oriented toward power systems [27], [28].

The above works, however, have not considered the impact of an adversary who is *actively* deciding which measurement and control signals reach their intended recipients. The considered packet dropping processes were *independent from the state of the system* and rather *static in nature* (e.g., Gaussian noise is maintained at a constant level over time). Clearly, an adversary does not need to abide by a particular process. On the contrary, a smart adversary will adapt his/her attack policy based on *the current state of the system and to influence future states* in order to maximize his/her reward subject to different costs.

The most related work to our particular instantiation on pheromone-based coordination methods is the work in [29], [30]. In [30], few heuristics were developed that aimed to inflict damage on the nodes through jamming a subset of the signals between the nodes. In [29], different hazards that can occur in robot swarms were investigated. One of those hazards is the complete failure of the communication modules. This had the effect of the nodes wandering off at random and not participating in the swarm. In contrast to the studies above, which were based on heuristics for the most part, the approach presented in this paper seeks to find efficient attack policies based on a solution to an optimization problem in an MDP framework that considers the overall state of the system and accounts for both the current and future impact of the attack actions.

In summary, our work differs from all previous work across three major dimensions. First, we consider application-driven attacks in sharp contrast to protocol-driven attacks. Second, the attacks are dynamic and sequential in nature as they are actively obtained through a mapping from the current state of the system to the action space to maximize the attacker's overall gain. Third, we consider a systematic approach based on a solution to an MDP, whereby optimal attack policies are designed to provide explicit performance guarantees.

III. The General Framework

In this section we present a unifying framework to model CPS with mobile nodes in the presence of adversaries.

A. A Unified Model of Mobile CPS with Adversaries

We consider a CPS composed of n mobile nodes deployed in a hostile environment populated by adversaries. We let $s_i(k)$ denote the state of node i at time interval k. The state of node i can be updated independently, based on the previous state $s_i(k-1)$ and the probability of a transition $p_i(s_i(k) = s_1|s_i(k-1) = s_0)$ as shown in Figure 1 (a). Alternatively, the state transition could also depend on control signals as shown in Figure 1 (b). We let $u_i(k)$ denote the control signal received by node i. Control signals can be received directly from a specific node (e.g., a base station in a centralized control), or from other nodes (distributed control). Each node has a local control component that updates its state in the absence of fresh control signals. A measurement signal, $y_i(k)$, is generated by node i and fed-back to the system at time interval k. Due to node mobility, the control and the measurement signals can only traverse valid network links. We let $L(k)$ indicate the set of direct links available at time interval k. These are calculated based on the locations of the nodes and the radio transmission ranges.
The goal is to evolve the system into an overall state $s(.)$ that meets particular functions (e.g., reaching a set of targets, covering an area, avoiding congested areas, etc...). The system state, $s(.) = \{s_i(.)\}_{i=1}^n$, is the joint states of all the nodes.

To account for adversaries, we let C denote the cost incurred in interfering with a given link[1]. Thus, at time interval k, the adversary has $2^{|L(k)|}$ options that range from not attacking any link to attacking all of them, where $|L|$ is used to denote the cardinality of a set L. We let $\hat{L}(k)$ denote the set of those links interfered with, where $\hat{L}(k) \subseteq L(k)$. The control and/or measurement signals traversing any link in the set $\hat{L}(k)$ are considered lost. To identify stealthy classes of attacks, we propose *reward functions* (Equation 1 below) to model the choice and impact of the adversarial policies in this environment, and the goal of the attacker is to solve the optimization problem in (2) over the choice of attack policies.

$$R(k) = D(s(k)) - C|\hat{L}(k)| \tag{1}$$

$$\max_{\mu_1,\mu_2,\dots} E\left[\sum_{k=1}^{\infty} \gamma^k R(k)|I_k\right] \tag{2}$$

To clarify, we let the attacker decide *when to interfere and which link(s) to interfere with.* Thus, (1) represents the reward, $R(k)$, obtained by the adversary at time k based on a chosen damage function D for driving the system into state $s(k)$, minus the cost incurred. The adversary aims to maximize the expected discounted cumulative reward over time by choosing attack policies (μ_1, μ_2, etc...) as shown in (2), where $\mu_k : \mathcal{I}_k \to \mathcal{A}_k$ is the policy at time interval k mapping the information state $I_k \in \mathcal{I}_k$ of the system at time k to an attack action $a_k \in \mathcal{A}_k$, and $0 < \gamma \leq 1$ is a discount factor. The discount factor ensures that the infinite summation in (2) is bounded. If $\gamma < 1$, then gains incurred far in the future have lesser weight than current gains. It is worth mentioning that other formulations are also admissible. In particular, if the attack policy is designed over a finite horizon, the infinite summation in (2) can be restricted to a summation over a finite interval T. Also, the case where $\gamma = 1$ is mathematically feasible when the system evolves to an absorbing cost-free termination state. In Section V we consider the latter scenario. Figure 1 (c) shows the effect of a single attack action, that moves the system into an inefficient state.

B. The Offense Component

To derive the optimal attack policy – for a given cost C and a damage function D – we iteratively solve the following Bellman equation [31]:

$$J(s) = \max_a \left\{ E[R(s,s',a)] + \gamma \sum_{s'} p(s'|s,a,u)J(s') \right\} \tag{3}$$

where $J(.)$ represents the value function, representing the value of being in state s. The first term on the RHS in (3) represents the immediate stage reward (Equation 1) and the second term is the future reward. $p(.)$ is the probability of a transition of the system to state s' from state s under the aggregate effect of the control signal u and the attack action a. The second term is multiplied by a discount factor γ, which is typically close to 1. Solving the equation above gives the optimal tradeoff between damage and cost from the standpoint of the adversary. Note that time dependence is irrelevant given the infinite horizon formulation in (2), which leads to a time invariant attack policy and a time-invariant Bellman equation as in (3).

Due to the large state space, solving the above equation may not be computationally feasible. Thus, we propose an approximate policy iteration method [32]–[34]. Exact policy iteration consists of 2 steps: policy evaluation and policy improvement. In the policy evaluation step, we start with an initial policy μ. Then, we solve a system of linear equations to evaluate the cost function $J_\mu(s)$ starting from state s and using policy μ:

$$J_\mu(s) = \sum_{s'} p(s'|s,\mu(s)) \left(R(s,s',\mu(s)) + \gamma J_\mu(s')\right) \tag{4}$$

where the summation is over the set of states s' that can be reached from state s and $R(s,s',\mu(s))$ is the reward obtained

[1]For ease of exposition we assume that C is fixed. However, the proposed framework naturally extends to the case where the cost C_ℓ is function of the attacked link $\ell \in L$. We relax this assumption in our evaluation.

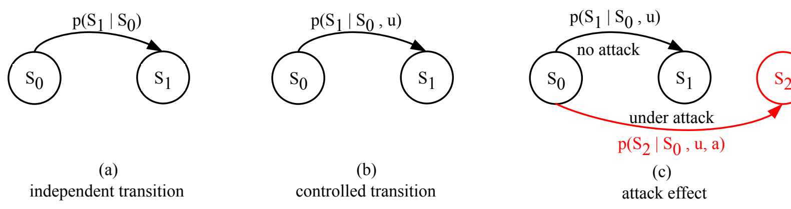

Fig. 1. Possible transitions from state S_0. (a) An independent transition from state S_0 to S_1. (b) A controlled transition from state S_0 to S_1 based on control u. (c) Attack effect causing a transition to an inefficient state S_2 based on the attack action a.

from the transition from s to every state in s' under policy $\mu(s)$. In the policy improvement step, an improved policy $\bar{\mu}$ is generated according to following equation:

$$\bar{\mu}(s) = \arg \max_{a \in \mathcal{A}(s)} \sum_{s'} p(s'|s,a) \left(R(s,s',a) + \gamma J_\mu(s')\right) \tag{5}$$

The improved policy is the one that maximizes the reward by selecting the best attack action a, from the set of actions $\mathcal{A}(s)$ available from state s. The improved policy $\bar{\mu}$ is then used as the new policy and a new iteration starts.

One of the main challenges with exact policy iteration is the size of the state space. If we let $|S|$ denote the size of the state space, and $|A|$ the cardinality of the attack space, the complexity of (4) is $O(|S|^3)$ (since it involves solving a system of $|S|$ linear equations), and the complexity of (5) is $O(|A||S|^2)$. For example, given a 5x5 grid with 6 nodes and 4 links, obtaining an optimal policy would require $O(|25^6|^3 + |2^4||25^6|^2)$ steps! Thus, an approximation becomes mandatory.

In the approximate variant of policy iteration, we approximate $J_\mu(s)$ with a parametric representation $\tilde{J}_r(s)$:

$$\tilde{J}_r(s) \quad = \quad \sum_{j=1}^{f} r_j \phi_j(s) \tag{6}$$

where $\phi = [\phi_1, \ldots, \phi_j, \ldots \phi_f]^T$ is a column of features, r is a row of weights (one for each feature), and f is the number of features. The idea is to extract f features that characterize the state s and approximate $J_\mu(s)$ by selecting r that solves a least square problem (between a Monte Carlo simulation and the current policy evaluation). The main difficulty is to judiciously select a representative set of features depending on the application. It is known that the linear combination of well chosen features can capture essential nonlinearities in the reward function [35]–[37].

IV. Attacks on Pheromone-based Coordination Methods

In this section, we instantiate the general framework described in the previous section for a mobile CPS application that uses pheromones for coordination between the mobile nodes.

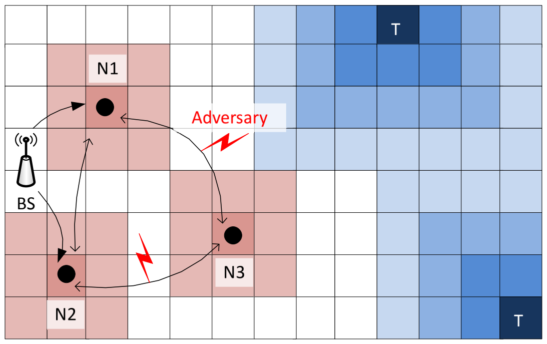

Fig. 2. A mobile CPS composed of 3 mobile nodes (N1, N2 and N3), a base station (BS) and two targets (T). Different color gradients indicate the pheromone levels.

A. Background

Pheromone-based coordination methods in Mobile CPS applications has three main characteristics: (1) pheromones can be pumped from a particular location (e.g., areas of interest) to attract mobile nodes, (2) pheromones evaporate at a constant rate, and (3) pheromone propagate to surrounding areas from where they are pumped. Mobile nodes independently decide their next move by computing movement probabilities based on the pheromones in the surrounding environment. In particular, nodes have low probabilities of moving into areas with repulsive pheromones and high probabilities of moving into areas with attractive pheromones. As a node moves, it withdraws all the positive (attractive) pheromones and deposits negative (repulsive) pheromones at its location to repel other nodes from the area, resulting in dispersive behavior. Both, attractive and repulsive pheromones evaporate over time, allowing the nodes to revisit an area once the pheromones have dissipated. Area of Interests (AOIs) are regions that periodically pump attractive pheromones in a location that diffuses to the surrounding area. These regions will be visited more often than the surrounding areas. A “pheromone map” denotes the nodes’s view of the pheromone levels in its environment. Figure 2 shows a Mobile CPS composed of 3 mobile nodes and two targets. The pheromone levels are depicted using different color gradients. The figure also shows a central base station (BS), which sends update information to some of the mobile nodes and whose exact role will become more clear later in this section.

Through adjusting the pheromone levels at particular areas in an environment, one can influence the movement probabilities and create different scenarios for various applications. For example, when all areas pump pheromones at a given level, the behavior of the nodes resembles covering an area equally. Alternatively if an attractive pheromone is pumped at specific locations, then the nodes would visit those locations more often than others. Similarly, repulsive pheromones can be placed in specific locations to indicate obstacles and barriers the nodes need to avoid (e.g., country borders). For more information about pheromone-based coordination methods, we refer the reader to references [18], [19].

B. The Model

Our model consists of discrete-time sequences in which each mobile node makes a decision based on its own version of the pheromone map. Each node maintains several pheromone maps that represent the state of the system: an internal map, maps for the other nodes (for example to avoid collision), and an AOI map. The internal map is always accurate because each node is continuously aware of its own location. The different maps maintained by a particular node are updated only when it receives a signal from the other nodes or when the BS sends an AOI update. The goal of the mobile nodes is to locate two targets in the shortest time possible without colliding with each other. The attack policy is to determine which signal or set of signals to interfere with in order to minimize the probability of detection and maximize the damage.

1) Three Pheromone Maps: Next, we describe the details of our proposed model. Let $A_i(k), i = 1, \ldots, n$ denote the internal map of node i at time k, where n is the total number of mobile nodes. This map represents the pheromone levels induced by the motion pattern of node i and the known evaporation, diffusion and deposit mechanisms described earlier. Hence, $A_i(k)$ is maintained and updated by, and perfectly known to, node i. In addition to its own map, each node keeps track of other nodes' maps. We let $W_{i,j}(k), i, j = 1 \ldots n, i \neq j$, denote the map of node j *as perceived and maintained by node* i at time k. These maps are based on the information exchanged between the nodes at different time steps. As we will describe later, it might be in the interest of the attacker to attack the exchange of these maps at particular time steps. Hence, these maps may be inaccurate and may not reflect an exact picture of the actual pheromone levels induced by the other nodes. Finally, each node also keeps an AOI map $T_i(k)$ for the targets and the area of interest. Again, this map which is exchanged between the nodes, is subject to attacks and hence not every node may have an exact version of the AOI map.

2) Motion Dynamics: At each time step, each node compiles a total pheromone map $M_i(k), i = 1, \ldots n$ for the whole field by aggregating the aforementioned maps $A_i(k), W_{i,j}(k)$ and $T_i(k)$. Specifically,

$$M_i(k) = T_i(k) + A_i(k) + \sum_{\substack{j=1 \\ j \neq i}}^{n} W_{i,j}(k), \tag{7}$$

where $T_i(k)$ is always represented by attractive pheromone levels, while $A_i(k)$ and $W_{i,j}(k)$ are represented by repulsive pheromone levels as nodes visit different locations. Each node decides its next move based on its own aggregate map M_i. In particular, the next move for node i is decided by creating a probability wheel based on the pheromone concentration in $M_i(k)$ at all possible reachable locations from the current location $X_i(k)$. The next position $X_i(k+1)$ for node i is obtained through a transition probability distribution and can be stochastically described through

$$X_i(k+1) \sim p(X_i(k+1) = x' | x_i(k) = x, M_i(k)). \tag{8}$$

where the notation $X \sim p$ denotes a random variable X with distribution p. For example, at each time step a node may be allowed to move up, down, left or right from its current location, and the probability of each future location is determined based on the pheromone concentration at the set of reachable locations as given by the aggregate map $M_i(k)$.

3) Potential Attacks and State Evolution: From the attacker's standpoint, the state of the system $s(k)$ at time k consists of the locations $X_i(k), i = 1, \ldots, n$, of the different nodes, in addition to their maintained maps $A_i(k), W_{i,j}(k)$ and $T_i(k), i = 1, \ldots, n$. To describe the state evolution, we consider the time evolution of every component in the state vector $s(k)$.

The new positions of the nodes follow the motion dynamics in (8). Note that $M_i(k)$ can be calculated from the actual state components according to (7).

The internal map $A_i(k+1)$ is updated by each node based on the actual internal map $A_i(k)$, the maps of the other nodes $W_{i,j}(k)$, the new position, the evaporation rate e of the pheromone, the constant value of the pheromone flavor d_i for the node deposits, and the diffusion rate p of the pheromone to the neighboring cells. As such, the internal map updates can be described through a stationary transformation f as

$$A_i(k+1) = f_i(A_i(k), W_{i,j}(k), n(k)) \tag{9}$$

where $n(k)$ is a spurious component that captures the randomness induced through the uncertainty of the motion dynamics. Note that $n(k)$ depends probabilistically only on the current state through the maps maintained by the nodes. The transformation f_i can be different for different nodes depending on the pheromone flavor d_i.

The update of the maps $W_{i,j}(k)$ is somewhat more involved as it depends on the attack action. To clarify, note that following each move the mobile nodes exchange their internal maps. Hence, if the attacker decides not to attack the message from j to i, then the i-th node obtains an exact description of j-th internal map. In this case,

$$W_{i,j}(k+1) = A_j(k+1) \tag{10}$$

However, if the message from j to i is attacked, the i-th node does not get the update and obtains $W_{i,j}(k+1)$ by simply decaying the pheromone level based on the known decay rate.

To describe the evolution of the AOI map T_i we need to consider multiple factors. First, in our model we assume that the targets are stationary. Also, the BS transmits the AOI map to the nearest node and we further assume that this particular signal cannot be attacked. Thus, one node always maintains the true pheromone level of the targets from the BS. Let i_0 denote this particular node receiving the update from the BS at time k. Node i_0 broadcasts the updated AOI map and this broadcast signal is subject to attack. Akin to the previous scenario, if this signal is not attacked, each node successfully receives the update, in which case $T_i(k+1)$ will represent the true AOI map. However, if this signal is attacked, $T_i(k+1)$ is updated by decaying the pheromone levels. Note that the update is triggered by a signal from the BS at a given time interval and hence the AOI map need not be updated at each step. In our current implementation, we consider an update every m intervals. In a different implementation, we assume that the AOI map is automatically updated without a trigger signal from the BS (c.f. Section V).

To summarize, the state evolution is captured through the motion dynamics (8), the map updates in (9), (10) and in the previous description. As such, the state evolution follows the standard MDP prescription in [38] according to a probabilistic law $p(s(k+1)|s(k), a(k))$, where $a(k) \in \mathcal{A}_k$ is the attack action and $\mathcal{A}_k$ is the available action space at time k. Note that $\mathcal{A}_k$ depends on k since the available control actions depend on whether an update of the AOI map is sent by the BS and also on the closest node during the update. The time dependence of the action space is shifted to the state by simply augmenting the state vector with a binary variable taking the value 1 every m intervals to designate that such states can be mapped to attack actions on the broadcast signal of the AOI map through the designed attack policy.

4) Distracting the Coordination Method: The goal of distracting the coordination between the nodes is to have them wander aimlessly by jamming a subset of signals between the nodes, thereby prolonging the duration of the mission or leading the nodes to collide with each other. Emulating their stealthy nature, we also associate a cost when an adversary decides to attack a signal. Hence the cost of an attack action depends on the cost of attacking a particular link, as well as the link(s) being attacked. This abstraction can also capture the time spent by an attacker to interfere with the communication over a given set of links.

In our implementation we consider a system with Q stationary targets. The problem terminates if all the targets are found or in case of collision. Hence, we augment our state space with an absorbing termination state. The damage function consists of three components: (i) some function of the minimum distance between the mobile nodes and the targets of interest; (ii) a reward for causing a collision; and (iii) a negative reward for the targets that have been found. For this particular setup, the damage function D is instantiated as

$$\begin{aligned} D(k) = &\sum_{i=1}^{n} \min_{j\in\{1,2,\ldots,Q\}} d(X_i(k), P_j) \\ &+ \alpha \sum_{i=1}^{n} \sum_{j\neq i} \mathbb{1}\{X_i(k) = X_j(k)\} \\ &- \beta \sum_{i=1}^{n} \sum_{j=1}^{Q} \mathbb{1}\{X_i(k) = P_j\} \end{aligned} \tag{11}$$

where $d(U, V)$ denotes the distance between U and V and P_j the location of (the stationary) target j. $\mathbb{1}\{\}$ is an indicator function, α the reward for a collision and β is the cost for a found target. The middle term in the summation in (11) corresponds to collision when any pairs of nodes end up in the same cell. The notion of collision does not necessarily translate into a physical accident between the nodes, but rather captures nodes getting into the proximity of each other leading to redundant work being done. The last term corresponds to a found target.

The goal is to maximize the total rewards over the choice of attack policies. Our problem is readily posed as a stochastic shortest path MDP with a termination state. Hence, the optimal attack policy is stationary and can be obtained as a solution to the Bellman equation described in (3). This policy is defined through a mapping, $\mu : \mathcal{S} \to \mathcal{A}$, from the state space to the action space.

5) Feature Selection and Approximate Policy Iteration: The optimal policy μ can be generally obtained using a standard technique such as policy or value iteration. However, in this context such an approach is computationally prohibitive considering the size of the state space. Instead, we resort to an approximate solution based on Approximate Policy Iteration [38] as described in Section III-B. The main difficulty lies in choosing an efficient set of representative features capturing the value of each state. To this end, we have selected the following set of features:

- Distance between the mobile nodes.
- Distance between the nodes and the AOIs.
- Flag indicating when the AOI map gets updated.
- Map difference of AOI maps between the nodes.
- Map difference between the nodes' pheromone maps and the truth map.

We remark that the online computational overhead of the proposed approach is minimal. In particular, the parametric representation in (6) is computed *offline* based on the simulation of independent trajectories starting from different representative states. In section V we provide a comprehensive performance evaluation of our proposed approach based on the selected features.

V. Performance Evaluation

In this section we report our evaluation of the Approximate Policy Iteration methods in identifying stealthy attacks on various systems based on the model outlined in Section IV.

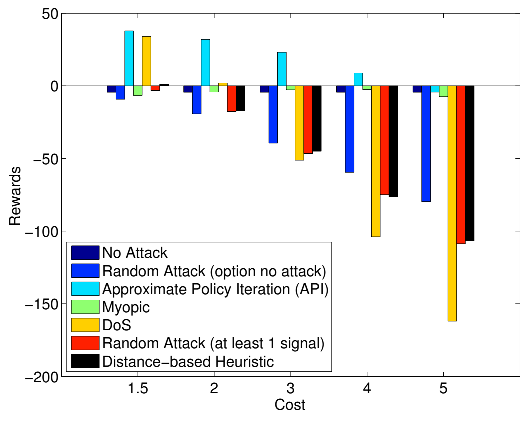

Fig. 3. System A: Rewards obtained for different attack costs under different policies.

A. System A

In this setup, we consider a system composed of two targets (e.g., AOI), two mobile nodes, and a base station (BS). The BS, which is stationary, sends the AOI map to the nearest node and is updated every 3 time steps. The area of operation consists of a 10x10 grid cells with few obstacles that the nodes must navigate around. Pheromone levels vary from -20 (to indicate a location that has just been visited by a node) to 20 (to indicate a location with a target). Pheromone levels decay by 1 unit every time step. From a particular location, their propagation decreases progressively by 2 units into the surrounding locations (e.g., 10, 8, 6, 4, 2, and 0).

The goal of this setup is for the mobile nodes to locate the two stationary targets in the least amount of time. Each mobile node decides its next move based on its local pheromone maps, the AOI map, and the maps of the other nodes as described in Section IV-B.2. Nodes have higher probability to move into cells with higher pheromone levels than those with lower level ones. The mapping of pheromone levels to probabilities was chosen to ensure that the targets are reached in the shortest amount of time when no attacks are mounted.

We consider an attacker who aims to distract the mobile nodes through interfering with a subset of signals exchanged between them. We assume that the attacker has full knowledge of the state of the system and that it incurs a cost of C when it attacks a given link. Thus, at any state, the attacker can choose between the following actions:

1) No attack with cost 0
2) Attack the signal from node 1 to node 2 with cost C
3) Attack the signal from node 2 to node 1 with cost C
4) Attack both signals with cost $2 \times C$

We instantiate the damage function D based on (11) where we set α to 500 and β to 100.

We start our API algorithm with 100 representative states that are chosen randomly from the state space. We use a myopic policy as a roll-out one in which the actions taken are based on the immediate rewards without considering the quality of the future state(s). From each representative state, we run 40 independent trajectories and we compute the average rewards across the trajectories. In each trajectory, the simulation ends once there is a collision or when both targets are located. A new policy is generated after each iteration and we track the weight vector r that produces the policy with the highest reward.

Once we obtain the weight vector, we compare our API policy against other policies on a completely different set of states that are also generated randomly. This ensures there is no intentional overlap between the training data and the ones we use for evaluation.

With the API method, there is no guarantee that the system will produce a better policy with each iteration as with exact policy iteration. Thus, we run the API method over a large number of iterations and choose the policy that produces the maximum reward. In many cases, we were able to find the best policy in the first 20 out of 100 iterations.

Figure 3 shows the rewards obtained for different attack costs and under different attack policies. In particular, we compare our API method to several other policies: (1) a no-attack policy that acts as a baseline, (2) a random attack policy in which an action is chosen uniformly at random, (3) a myopic attack in which only the immediate reward is used in selecting an action without regard to the future rewards, (4) a DoS attack in which both links are attacked, (5) a random attack in which at least one link is attacked, and (6) a distance-based heuristic in which both links are attacked if a node gets within a certain distance to a target. We chose a distance of 3 cells since it provided the best case for this heuristic for comparison. We only show the interesting region based on the attack costs. If the cost of the attack is very low, API matches a DoS attack and if the cost is very high, API matches a no-attack policy. One can see that API was able to identify potent attacks that can perform significantly better than other attack policies from the standpoint of the attacker.

Attack Cost	1.5	2	3	4	5
No-attack	67.3%	71.4%	74.5%	84.8%	89.4%
One signal attacked	29.8%	27.4%	25%	15.1%	10.6%
Both signals attacked	2.9%	1.2%	0.5%	0.1%	0%

TABLE I
DISTRIBUTION OF ACTIONS TAKEN BY API AS WE VARY THE ATTACK COST FOR SYSTEM A.

Table I shows the distribution of the control actions taken by API as the cost of the attack is varied. One can see that API judiciously adapts the actions based on the cost – becoming less aggressive as the cost of the attack is increased. Correlating Table I with Figure 3, one can observe that at the lowest cost, API was performing similar to a DoS attack, but with a fewer aggressive attack actions (no-attack action was chosen 67% of the time).

Figure 4 shows the percentage of increase in time to locate the targets for different costs under all policies in comparison to the no-attack policy (under which the targets are located in the shortest time). One can observe that API causes about

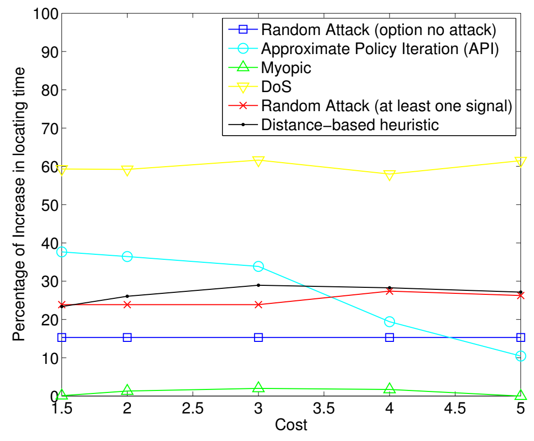

Fig. 4. Increase in time to locate the targets for different costs and under different policies. Results are presented for System A.

40% increase in the time to locate the targets at lower costs and about 10% increase at higher costs. Note that a lower value does not imply a lower reward. Indeed, while the API policy achieves the highest reward (as seen previously in Fig. 3), the curve corresponding to the API policy in Fig. 4 sits below that of random attack policies at the high cost regime. This should come at no surprise since the API policy adapts both the attack level and its timing in order to achieve a favorable tradeoff between the damage and the cost, and thus a higher total reward. As such, Fig. 4 underscores the adaptivity of API, which is a key feature of the proposed policy in sharp contrast to all the other policies.

B. System B

In System A we assumed that the attacker is always successful when mounting an attack. System B considers the case when the attacker is only successful with a certain probability. Thus, in System B, the cost of the attack is incurred whenever the attacker decides to attack, but the attack is only effective a certain percentage of time.

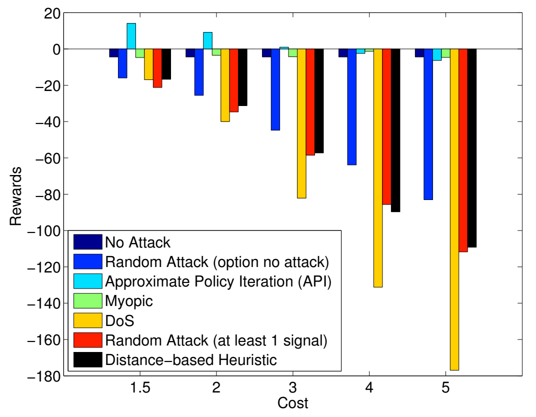

Fig. 5. System B: Rewards obtained for different attack costs under different policies with attack success rate of 75%.

Figure 5 shows similar results to those presented for System A but for an attack that is successful 75% of the time. One can see again that API outperforms all other policies (except at the high-end cost of 5). As mentioned earlier, it is not always guaranteed that API will converge to a better policy. Moreover, at high cost, the penalty of incurring the cost without actually causing any damage becomes higher.

C. System C

System C is a variant of System A in which we assign different attack costs for attacking different links between the nodes. This scenario captures different aspects such as: (1) the attacker maybe closer to one of the nodes and thus easier for him/her to jam a particular link, or (2) the attacker is aware of a particular vulnerability on a specific link that is cheaper to exploit. For System C, we assume the attacker can decide between the following actions:

1) No attack with cost 0
2) Attack the signal from node 1 to node 2 with cost C_{12}
3) Attack the signal from node 2 to node 1 with cost C_{21}
4) Attack both signals with cost $C_{12} + C_{21}$

We study two variants; In System C-1, we set C_{12} and C_{21} to 2 and 4, respectively. In System C-2, we set C_{12} and C_{21} to 1 and 5, respectively. These two systems can also be put in comparison to System A with a cost of 3, since the overall cost in all these systems is 6.

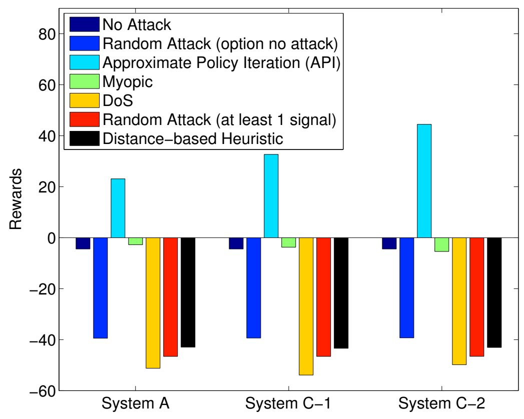

Fig. 6. Systems C-1 ($C_{12} = 2$ and $C_{21} = 4$) and C-2 ($C_{12} = 1$ and $C_{21} = 5$) in comparison to System A ($C_{12} = 3$ and $C_{21} = 3$).

Figure 6 shows the rewards obtained for System C-1 and C-2 in comparison to System A. Once again API outperforms other policies. Notice also how the rewards get higher as the degree of imbalance between the costs in attacking the links increases; API was able to capitalize on the weakest link more often to maximize the effectiveness of the attacks. This is more evident if we look at the distribution of actions taken. For System C-1, the distribution was 74.9%, 22.6%, 2.4% and 0.1%, for actions 1, 2, 3 and 4, respectively, whereas for System C-2, it was 67%, 31.4%, 1.4% and 0.2%, for actions 1, 2, 3 and 4, respectively.

VI. Conclusions

There has been a large number of frameworks that capitalize on cooperation between mobile nodes over wireless links to enable new mobile CPS applications. The security of these systems against jamming attacks becomes a critical issue as our reliance on these systems continue to grow. To this end, we have proposed a framework that enables the identification of different classes of stealthy attacks in a systematic approach. We have identified vulnerabilities in one of the common pheromone-based coordination mechanisms used in Mobile CPS applications. We have shown that by jamming a small subset of the links, an attacker can distract the nodes leading to repetitive tasks being done and/or nodes getting into the vicinity of each other, possibly leading to collisions. Our results show that the developed attack policies can adaptively adjust their strength depending on the system setup. It is shown that the attack policy approaches the DoS policy in the low cost regime and the no-attack policy in the high cost regime and in between many classes of stealthy policies exist. The exposed attack policies outperform other policies such as myopic, random and DoS attacks. We believe this work is the first that looks into identifying attacks against emerging mobile CPS.

Acknowledgement

This work was supported in part by NSF awards CNS-1149397 and CCF-1320547.

References

[1] M. Guirguis and G. Atia, "Stuck in Traffic (SiT) Attacks: A Framework for Identifying Stealthy Attacks that Cause Traffic Congestion," in *Proceedings of the 77th IEEE Vehicular Technology Conference (VTC)*, Dresden, Germany, June 2013.

[2] J. Hespanha, P. Naghshtabrizi, and Y. Xu, "A Survey of Recent Results in Networked Control Systems," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 138–162, 2007.

[3] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, M. Jordan, and S. Sastry, "Kalman Filtering with Intermittent Observations," *IEEE Transactions on Automatic Control*, vol. 49, no. 9, pp. 1453–1464, 2004.

[4] S. Smith and P. Seiler, "Estimation with Lossy Measurements: Jump Estimators for Jump Systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 12, 2003.

[5] M. Yu, L. Wang, T. Chu, and G. Xie, "Stabilization of Networked Control Systems with Data Packet Dropout and Network Delays via Switching System Approach," in *Proceedings of the IEEE Conference on Decision and Control*, 2004.

[6] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. Sastry, "Foundations of Control and Estimation Over Lossy Networks," *IEEE*, vol. 95, no. 1, p. 163, 2007.

[7] C. Hadjicostis and R. Touri, "Feedback Control Utilizing Packet Dropping Network Links," in *Proceedings of IEEE Conference on Decision and Control*, Las Vegas, NV, December 2002.

[8] J. Nillson, "Real-time control systems with delays," Ph.D. Thesis, Lund Institute of Technology, Lund, Sweden, 1998.

[9] M. Huang and S. Dey, "Stability of Kalman Filtering with Markovian Packet Losses," *Automatica*, vol. 43, no. 4, pp. 598–607, 2007.

[10] W. Xu, K. Ma, T. Wade, and Y. Zhange, "Jamming Sensor Networks: Attacks and Defense Strategies," *IEEE Network*, 2006.

[11] D. Thuente and M. Acharya, "Intelligent Jamming in Wireless Networks with Applications to 802.11b and other Networks," in *Proceedings of IEEE MILCOM*, Washington, DC, October 2006.

[12] L. Sang and A. Arora, "Capabilities of Low-power Wireless Jammers," in *Proceedings of INFOCOM*, Rio de Janeiro, Brazil, April 2009.

[13] K. Pelechrinis, M. Iliofotou, and V. Krishnamurthy, "Denial of Service Attacks in Wireless Networks: The Case of Jammers," *IEEE Communications Surveys & Tutorials*, vol. 13, no. 2, 2011.

[14] Y. Law, M. Palaniswami, L. Hoesel, J. Doumen, P. Hartel, and P. Havinga, "Energy-Efficient Link-Layer Jamming Attacks gainst Wireless Sensor Network MAC Protocols," *ACM Transactions on Sensor Networks (TOSN)*, vol. 5, no. 1, pp. 1–38, 2009.

[15] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The Feasibility of Launching and Detecting Jamming Attacks in Wireless Networks," in *Proceedings of ACM MobiHoc*, Urbana-Champaign, IL, May 2005.

[16] B. Awerbuch, A. Richa, and C. Scheideler, "A Jamming-resistant MAC Protocol for Single-hop Wireless Networks," in *Proceedings of ACM PODC*, Toronto, Canada, August 2008.

[17] J. Chiang and Y. Hu, "Cross-layer Jamming Detection and Mitigation in Wireless Broadcast Networks," *IEEE/ACM Transactions on Networking (TON)*, vol. 19, no. 1, pp. 286–298, 2011.

[18] J. Sauter, R. Matthews, H. Parunak, and S. Brueckner, "Performance of Digital Pheromones for Swarming Vehicle Control," in *Proceedings of the Fourth International joint Conference on Autonomous Agents and Multiagent Systems*. ACM, 2005, pp. 903–910.

[19] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence: From Natural to Artificial Systems*. Oxford University Press, USA, 1999.

[20] T. Yang, "Networked Control System: A Brief Survey," *IEE Proceedings Control Theory and Applications*, vol. 153, no. 4, 2006.

[21] US-CERT, "Control Systems Security Program (CSSP)," http://www.us-cert.gov/control_systems/.

[22] US-GAO, "Critical Infrastructure Protection. Multiple Efforts to Secure Control Systems are Under Way, but Challenges Remain," http://www.gao.gov/new.items/d071036.pdf.

[23] A. Cárdenas, S. Amin, and S. Sastry, "Research Challenges for the Security of control Systems," in *Proceedings of the 3rd USENIX Workshop on Hot Topics in Security*, San Jose, CA, July 2008.

[24] ——, "Secure Control: Towards Survivable Cyber-Physical Systems," in *Proceedings of the Internatinal Conference on Distributed Computing Systems (ICDCS)*, Beijing, China, June 2008.

[25] S. Amin, A. Cárdenas, and S. Sastry, "Safe and Secure Networked Control Systems under Denial-of-Service Attacks," *Hybrid Systems: Computation and Control*, pp. 31–45, 2009.

[26] A. Cárdenas, S. Amin, Z. Lin, Y. Huang, C. Huang, and S. Sastry, "Attacks against Process Control Systems: Risk assessment, Detection, and Response," in *Proceedings of the 6th ACM Symposium on Information, Computer and Communications Security*. ACM, 2011, pp. 355–366.

[27] A. Teixeira, S. Amin, H. Sandberg, K. Johansson, and S. Sastry, "Cyber Security Analysis of State Estimators in Electric Power Systems," in *Proceedings of the IEEE CDC*. IEEE, 2010.

[28] A. Teixeira, H. Sandberg, and K. Johansson, "Networked Control Systems under Cyber Attacks with Applications to Power Networks," in *American Control Conference (ACC), 2010*. IEEE, 2010.

[29] A. Winfield and J. Nembrini, "Safety in Numbers: Fault-tolerance in Robot Swarms," *International Journal of Modelling, Identification and Control*, vol. 1, no. 1, pp. 30–37, 2006.

[30] J. Kelly, S. Richter, and M. Guirguis, "Stealthy Attacks on Pheromone Swarming," in *Proceedings of IEEE CogSIMA*, San Diego, CA, February 2013.

[31] R. Bellman, "A Markovian Decision Process," *Journal of Mathematics and Mechanics*, vol. 6, no. 4, 1957.

[32] L. Getoor and B. Taskar, "Introduction to Statistical Relational Learning," The MIT Press, 2007.

[33] D. Bertsekas, "Approximate Policy Iteration: A Survey and Some New Methods," *Journal of Control Theory and Applications*, 2011.

[34] R. Sutton and A. Barto, "Reinforcement Learning," MIT Press, Volume 18, 1988.

[35] L. Busoniu, D. Ernst, B. De Schutter, and R. Babuska, "Cross-Entropy Optimization of Control Policies with Adaptive Basis Functions," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 41, no. 1, 2011.

[36] D. Di Castro and S. Mannor, "Adaptive Bases for Reinforcement Learning," *Machine Learning and Knowledge Discovery in Databases*, pp. 312–327, 2010.

[37] H. Yu and D. Bertsekas, "Basis function adaptation methods for cost approximation in mdp," in *IEEE Symposium on Adaptive Dynamic Programming and Reinforcement Learning*, Nashville, TN, April 2009.

[38] D. Bertsekas and J. Tsitsiklis, *Neuro-Dynamic Programming*. Athena Scientific, 1996.