Deceitful Attacks in Security Games

Thanh H. Nguyen, Michael P. Wellman, Arunesh Sinha
University of Michigan, Ann Arbor
{thanhhng,wellman,arunesh}@umich.edu

Abstract

Given recent applications of defender-attacker Stackelberg Security Games in real-world domains such as wildlife protection, a majority of research has focused on addressing uncertainties regarding the attacker in these games based on the exploitation of attack data. However, there is an important challenge of deceitful attacks; the attacker can manipulate his attacks to mislead the defender, leading her to conduct ineffective patrolling strategies. In this work, we focus on addressing this challenge while providing the following main contributions. First, we introduce a new game model with uncertainty about the attacker type and repeated interactions between the players. In our game model, the defender attempts to collect attack data over time to learn about the attacker type while the attacker aims at playing deceitfully. Second, based on the new game model, we propose new game-theoretic algorithms to compute optimal strategies for both players. Third, we present preliminary experiment results to evaluate our proposed algorithms, showing that our defense solutions can effectively address deceitful attacks.

Introduction

Defender-attacker Stackelberg Security Games (SSG) have been successfully applied for solving many real-world security problems (Tambe 2011; Fang et al. 2016; Basilico, Gatti, and Amigoni 2009; Letchford and Vorobeychik 2011). In these security problems, there exist different uncertainties regarding the attacker such as uncertainties in his types, preferences, and behavior. This leads to a challenging research question: How to determine effective patrolling strategies in security scenarios with such uncertainties? Fortunately, in security domains such as wildlife protection (Fang et al. 2016), repeated interactions between rangers and poachers in conservation areas allow rangers to collect poaching signs over time. Thus, the rangers can exploit the poaching data to acquire knowledge about the poachers, specifically learning aforementioned unknown properties of the poachers. The rangers can then plan patrolling strategies accordingly.

However, the defender has to face with another important challenge: the attacker can manipulate his attacks in order to deceive the defender. Deceitful attacks could significantly deteriorate the learning outcome for the defender, thus resulting in ineffective patrolling strategies. Therefore, the defender must take into account the attacker's manipulation of attacks when using the collected attack data to learn about the attacker. In fact, existing work on uncertainties in SSG assumes the attacker is always truthful (Haghtalab et al. 2016; Balcan et al. 2015; Blum, Haghtalab, and Procaccia 2014; Nguyen et al. 2016; Kar et al. 2017; Gholami et al. 2017). Therefore, their proposed patrolling solutions based on collected attack data are vulnerable to deceitful attacks. In this work, we focus on addressing the challenge of deceitful attacks in a repeated SSG setting with uncertainty in the attacker's type. We aim at (i) studying the attacker's manipulation of attacks to mislead the defender about his type; and (ii) determining effective patrolling strategies given the attacker's manipulation.

In particular, we provide the following main contributions. First, we introduce a new repeated security game model with unknown attacker type. In our game model, the defender attempts to collect attack data over multiple time steps to learn about the attacker's type and then plan patrolling strategy accordingly. On the other side, the attacker tries to fool the defender about his type by manipulating his attacks at every time step. In addition, the attacker's payoffs dynamically change over time, adding further uncertainty which needs to be taken into account in the defender's strategic reasoning. To that end, our model provides a hierarchical view of strategic reasoning of players regarding their assumptions about the opponent's strategies.

Second, we propose new game-theoretic algorithms to (i) compute optimal deceitful attack strategies; and (ii) compute optimal patrolling strategies given the manipulated attack data. Our algorithms take into account future possibilities of player's interactions to recursively reason about players' strategies at every time step. In addition, we propose a limited look-ahead heuristic to limit the number of future time steps to consider in computing players' strategies. This heuristic allows us to overcome the computational challenge of exploring exponentially many future possibilities. Third, we provide preliminary experiment results, analyzing our game model over proposed attack and defense solutions.

Related Work

Learning in Security Games. Previous work on SSG studies the challenge of uncertainties in these games in a re-

peated game setting in which the defender can collect attack data over multiple time steps (Haghtalab et al. 2016; Balcan et al. 2015; Blum, Haghtalab, and Procaccia 2014; Nguyen et al. 2016; Kar et al. 2017; Gholami et al. 2017). The collected data can be used to learn a specific unknown property of the attacker, which is then exploited to find effective patrolling strategies. Previous work heavily relies on the assumption that the attacker plays truthfully at every time step. Therefore, their proposed defense solutions are vulnerable to deceitful attacks. In our work, we study the challenge of uncertainty in the attacker's type in a repeated Stackelberg security game in which the attacker aims at manipulating his attacks to mislead the defender about his type.

Secrecy and Deception in Security Games. The problem of secrecy and deception has been widely studied in security games (Guo et al. 2017; Bier and Zhuang 2008; Rabinovich et al. 2015; Farrell and Rabin 1996; Brown et al. 2005; Hendricks and McAfee 2006; Xu et al. 2015). Previous work studies security scenarios in which information available to the defender and the attacker is asymmetric. In previous work, the defender can exploit such asymmetry property by strategically revealing or disguising her information to the attacker. This results in responses of the attacker which are in favor of the defender. For example, in (Guo et al. 2017), the defender can strategically disguise her defense resources to deceive the attacker about the defender's type. In this work, we study an opposite scenario in which the attacker acts deceitfully to mislead the defender.

Adversarial Machine Learning. Machine learning is an important tool in security research to analyze and identify malicious activities in different security scenarios such as intrusion detection and spam email filtering. However, machine learning in an adversarial environment is faced with an important challenge that opponents may try to cause machine learning algorithms to fail in many ways. Recently, there have been several studies on adversarial machine learning, attempting to investigate different attack scenarios on machine learning algorithms (Brückner and Scheffer 2011; Lowd and Meek 2005; Barreno et al. 2006; Brückner, Kanzow, and Scheffer 2012; Barreno et al. 2010). For example, *causative* attacks alters the training process by influencing the training data or *exploratory* attacks attempts to discover information about the learner and its training data. Different machine learning algorithms are then proposed which can resist these sophisticated attacks. In this work, we focus on a causative attack scenario in security games. Existing work on causative attacks in adversarial learning relies on prediction accuracy of the learner as the direct measure to study strategic solutions for both players. Our work, on the other hand, aims at obtaining effective patrolling strategies which can minimize the damage of attacks in security games, given some learning result based on collected attack data.

Background

In Stackelberg Security Games (SSG), a defender attempts to allocate limited security resources to protect a set of important targets, $\mathbf{N} = \{1, 2, \ldots, N\}$ (Tambe 2011). On the other side, an attacker aims to attack one of these targets. Suppose the defender has $K < N$ resources. A *pure* defense strategy is an allocation of these K resources over the targets, each protecting one target. A *mixed* defense strategy is a probability distribution over pure defense strategies. Similarly, a *pure* attack strategy is to attack a target in $\mathbf{N}$. A *mixed* attack strategy is a probabilistic distribution over pure attack strategies. In the Stackelberg framework, the defender commits to a mixed strategy. The attacker is aware of the defender's mixed strategy and thus, he can play a best response (i.e., best pure attack strategy) accordingly.

The defender's mixed strategy can be equivalently represented as marginal coverage probabilities over the targets. Let x_i denote the probability the defender protects target $i \in \mathbf{N}$. A mixed defense strategy $\mathbf{x} = \{x_1, \ldots, x_N\}$, $x_i \in [0, 1]$, is *feasible* if $\sum_{i \in \mathbf{N}} x_i \leq K$. We denote by $\mathbf{X}$ the set of feasible defense strategies. Let $\mathbf{y} = \{y_1, y_2, \ldots, y_N\}$, $y_i \in [0, 1]$, $\sum_{i \in \mathbf{N}} y_i = 1$, denote a feasible mixed attack strategy. $\mathbf{Y}$ denotes the set of all feasible attack strategies.

If the attacker attacks a target $i \in \mathbf{N}$, he obtains a reward $R_i^a > 0$ if the defender is not protecting this target. Otherwise, the attacker receives a penalty $P_i^a < 0$. Conversely, the defender gets a penalty $P_i^d < 0$ in the former case and a reward $R_i^d > 0$ in the later case. We denote by $(\mathbf{R}^a, \mathbf{P}^a) = \{(R_i^a, P_i^a)\}$ and $(\mathbf{R}^d, \mathbf{P}^d) = \{(R_i^d, P_i^d)\}$ where $i \in \mathbf{N}$ the payoffs of the attacker and the defender respectively. Given a mixed defense strategy $\mathbf{x}$, an attack of target i yields an expected utility for defender and attacker respectively, which is computed as follows:

$$U_i^d(\mathbf{x}) = x_i R_i^d + (1 - x_i) P_i^d,$$
$$U_i^a(\mathbf{x}) = x_i P_i^a + (1 - x_i) R_i^a.$$

Game Model

Repeated security game

We consider a repeated Stackelberg security game over a time horizon $\mathbf{T} = \{1, 2, \ldots, T\}$. At each time step $t \in \mathbf{T}$, the defender commits to a mixed strategy $\mathbf{x}^t \in \mathbf{X}$, the attacker responds by playing an attack strategy $\mathbf{y}^t \in \mathbf{Y}$. In our game, there is a discrete set of attacker types $\mathbf{\Lambda} = \{\lambda\}$ associated with a prior distribution $\mathbf{p} = \{p^\lambda\}$ — p^λ is the probability the attacker is of type λ where $\sum_\lambda p^\lambda = 1, p^\lambda \in (0, 1)$. At the beginning of the game, nature draws an attacker type λ from the distribution $\mathbf{p}$. The attacker is aware of his type λ. On the other hand, the defender is aware of the prior distribution of attacker types $\mathbf{p}$ but she does not know the drawn attacker type.

Furthermore, for each attacker type $\lambda \in \mathbf{\Lambda}$, the attacker's payoffs at each time step $t \in \mathbf{T}$ is governed by a random environmental factor. The value of this factor varies over time, resulting in different payoff realizations for the attacker across different time steps. For example, in wildlife protection, poachers' payoffs depend on various features such as available poaching resources and animal density which change over time. Formally, each attacker type is associated with a payoff distribution f^λ over a continuous payoff space, denoted by $\mathbf{\Omega}^\lambda$. At each time step t, a payoff realization of the attacker type λ, denoted by $(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) \in \mathbf{\Omega}^\lambda$, is randomly drawn by nature from the distribution f^λ. We assume

that both players know this payoff distribution, f^{λ}, for each type $\lambda \in \mathbf{\Lambda}$. In our game, the realizations of the attacker's payoffs are only revealed to the attacker at the beginning of each time step. On the other hand, these realizations at all time steps are unknown to the defender.

Players' Strategy

Defense strategy: At each t, the defender observes the attacked target z^t which is randomly drawn from the attack strategy $\mathbf{y}^t$. The defender can collect attack data $\{(\mathbf{x}^t, z^t)\}$ over multiple time steps to predict which attacker type is playing. Based on the prediction result, she then decides on which patrolling strategies to execute in future time steps. In our game, we consider the scenario in which the defender's patrolling plan over $\mathbf{T}$ consists of two separate phases:

- **Learning phase:** The defender plans ahead of time a set of different defense strategies to play in the first T^l time steps. The main goal of this phase is to collect attack responses to learn the attacker type. We assume the attacker is aware of this set of defense strategies in advance.
- **Execution phase:** Based on the learning result, the defender decides on optimal defense strategies in next $T^e = T - T^l$ time steps. We assume that in this execution phase, the defender commits to the same optimal defense strategy with respect to the learning result at every time step.

Attack strategy: At each time step, given the attacker's payoff realization and the defender's strategy, the attacker can always play a best response (i.e., attack a target with the highest expected utility). However, he can intentionally deviate from best responses in the learning phase, which could mislead the defender about his type. Consequently, the defender may choose a patrolling strategy in the execution phase which is in favor of the attacker. Essentially, the attacker only has to decide on which attack strategies to play in the learning phase to mislead the defender. He then can always play a best response against the patrolling strategy chosen by the defender in the execution phase.

Strategic reasoning

Given the game setting and the players' strategy space, there are two important questions: (i) How should the attacker choose which attack strategy to play at each time step in the learning phase in order to mislead the defender about his own type and benefit the most? (ii) How should the defender exploit the attack data to decide on the defense strategy in the execution phase, given that the attacker is trying to mislead her? In fact, the answer for (i) depends on what assumption the attacker makes about the defender's exploitation of attack data. Likewise, the answer for (ii) relies on what assumption the defender has with respect to the attacker's manipulation of attacks in the learning phase. Essentially, the strategic reasoning for both players can be described via a hierarchical view as follows:

- Level-0: The level-0 attacker always plays a best response at every time step. The level-0 defender predicts the attacker type based on the collected attack data in the learning phase, assuming the attacker is at level-0. Then the defender chooses an optimal defense strategy to play in the execution phase based on the learning result.
- Level-1: The level-1 attacker intentionally plays attack strategies (which may be different from the best responses) in the learning phase to mislead the defender about his type, assuming the defender is at level-0. On the other hand, the defender follows the same procedure as at level-0 but assumes the attacker is at level-1.
- Level-L ($L > 1$): Both players follow the same strategic procedure as at level-1. Yet, the attacker assumes the defender is at level $L-1$ while the defender assumes the attacker is at level L.

In this work, we focus on the level-0 and level-1 strategic reasoning. We aim at (i) computing optimal attack strategies for the attacker in the learning phase; and (ii) computing an optimal defense strategy in the execution phase, given the attack data collected in the learning phase.

Level-0

At level-0, the attacker always plays a best response at every time step. Therefore, given the attack data collected in the learning phase $\{(\mathbf{x}^t, z^t)\}$ where $t \in \{1, 2, \ldots, T^l\}$, the defender can compute the posteriori distribution of attacker types based on Bayes' rule as follows:

$$\begin{aligned} p(\lambda \mid (\mathbf{x}^1, z^1), (\mathbf{x}^2, z^2), \ldots, (\mathbf{x}^{T^l}, z^{T^l})) \\ &\propto p^{\lambda} \prod_t p(z^t \mid \mathbf{x}^t, \lambda) \\ &= p^{\lambda} \prod_t \int_{\triangle(\mathbf{x}^t, z^t)} f^{\lambda}(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) d(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) \end{aligned}$$

where the payoff set $\triangle(\mathbf{x}^t, z^t) = \{(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) \in \mathbf{\Omega}^{\lambda} \mid z^t \text{ is a best response w.r.t. } (\mathbf{x}^t, \mathbf{R}^{a,t}, \mathbf{P}^{a,t})\}$. Given the posteriori distribution, the defender has to decide on which patrolling strategy to execute in the execution phase. We consider three approaches:

MAP-based approach

The defender uses the Maxiumum A Posteriori (MAP) estimation to select the attacker type with the highest posteriori probability (that is a point estimate):

$$\lambda^* = \operatorname*{argmax}_{\lambda \in \mathbf{\Lambda}} p(\lambda \mid (\mathbf{x}^1, z^1), (\mathbf{x}^2, z^2), \ldots, (\mathbf{x}^{T^l}, z^{T^l}))$$

Then she aims at computing an optimal defense strategy, $\mathbf{x}^*$, against the type λ^* to play in the execution phase. Essentially, $\mathbf{x}^*$ is computed as to maximize the defender's expected utility at each time step $t \in \{T^l + 1, T^l + 2, \ldots T\}$ in the execution phase, which can be formulated as follows:

$$\begin{aligned} \mathbf{x}^* = \operatorname*{argmax}_{\mathbf{x} \in \mathbf{X}} \int_{(\mathbf{R}^a, \mathbf{P}^a) \in \mathbf{\Omega}^{\lambda^*}} \Big[f^{\lambda^*}(\mathbf{R}^a, \mathbf{P}^a) \\ \times U^d_{i^*(\mathbf{x}, \mathbf{R}^a, \mathbf{P}^a)}(\mathbf{x}) \Big] d(\mathbf{R}^a, \mathbf{P}^a) \end{aligned}$$

where $i^*(\mathbf{x}, \mathbf{R}^a, \mathbf{P}^a) = \operatorname{argmax}_{i \in \mathbf{N}} U^a_i(\mathbf{x}, (\mathbf{R}^a, \mathbf{P}^a))$ is the target with highest expected utility for the attacker.

Practically, this optimization problem can be approximately solved by discretizing the continuous payoff space $\mathbf{\Omega}^{\lambda^*}$ and then applying existing algorithms for solving Bayesian security games (Tambe 2011).

Bayesian approach

The defender aims at computing an optimal defense strategy, $\mathbf{x}^*$, against the posteriori distribution over all attacker types in $\mathbf{\Lambda}$. This can be formulated as an optimization problem of maximizing the defender's expected utility at each time step $t \in \{T^l+1, T^l+2, \dots T\}$ in the execution phase, which is formulated as follows:

$$\mathbf{x}^* = \underset{\mathbf{x}\in\mathbf{X}}{\operatorname{argmax}} \sum_{\lambda\in\mathbf{\Lambda}} \Big[p(\lambda \mid (\mathbf{x}^1, z^1), (\mathbf{x}^2, z^2), \dots, (\mathbf{x}^{T^l}, z^{T^l})) \times \int_{(\mathbf{R}^a,\mathbf{P}^a)\in\mathbf{\Omega}^\lambda} f^\lambda(\mathbf{R}^a, \mathbf{P}^a) U^d_{i^*(\mathbf{x},\mathbf{R}^a,\mathbf{P}^a)}(\mathbf{x}) d(\mathbf{R}^a, \mathbf{P}^a)\Big]$$

Similarly, we can discretize the continuous payoff space $\mathbf{\Omega}^\lambda$ for all attacker types $\lambda \in \mathbf{\Lambda}$ and apply Bayesian-game algorithms to approximately solve this optimization problem.

No learning approach

The defender aims at computing an optimal defense strategy $\mathbf{x}^*$ with respect to the prior distribution of attacker types, $\mathbf{p}$, over $\mathbf{\Lambda}$. In other words, she does not exploit the attack data collected in the learning phase. The problem of finding $\mathbf{x}^*$ in this case is similar to the posteriori-based approach but with respect to the prior distribution $\mathbf{p}$. This prior-based strategy is not influenced by the attacker strategies in the learning phase. We consider this defense strategy as a baseline.

Level-1

At level-1, the attacker determines which attack strategy to play at each time step in the learning phase to mislead the defender about his type, assuming the defender is at level-0. On the other hand, the defender has to decide which defense strategy to execute in the execution phase given the collected attack data $\{(\mathbf{x}^\mathbf{t}, z^t)\}$, assuming the attacker is at level-1.

Optimal attack strategies

In the execution phase, the attacker only needs to always play a best response at each time step. Therefore, the attacker's main goal is to decide attack strategies in the learning phase. We denote by $\hat{\lambda}$ is the true attacker type. At each time step t, the payoff realization $(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) \in \mathbf{\Omega}^{\hat{\lambda}}$, which is drawn from the distribution $f^{\hat{\lambda}}$, is revealed to the attacker. However, he is uncertain about future payoff realizations. Therefore, the attacker has to take into account all possible payoff realizations and corresponding attack strategies in future time steps to make decision on which attack strategy to play at time step t. In fact, finding optimal attack strategies in the learning phase can be solved recursively as follows:

At last time step T^l. At time step T^l, the attacker's payoff realization $(\mathbf{R}^{a,T^l}, \mathbf{P}^{a,T^l})$ is revealed to the attacker. The attacker's goal is to find an optimal attack strategy at T^l such that his accumulated utility in expectation over all time steps $\{T^l, T^l+1, \dots, T\}$ is maximized. We denote the attacker's actions at previous time steps $t \in \{1, 2, \dots, T^l-1\}$ by $\mathbf{z}^{T^l-1} = \{z^1, z^2, \dots, z^{T^l-1}\}$. Note that these actions are randomly drawn from his attack strategies $\{\mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^{T^l-1}\}$ and are known to the defender. The defense strategy chosen in the execution phase only depends on the pure attack actions played in the learning phase, and not on the mixed attack strategies. Therefore, finding an optimal attack strategy at T^l can be formulated as the following maximization problem:

$$Q(\mathbf{z}^{T^l-1}, \mathbf{R}^{a,T^l}, \mathbf{P}^{a,T^l}) = \max_{\mathbf{y}\in\mathbf{Y}} \sum_{i\in\mathbf{N}} y_i \Big[U^a_i(\mathbf{x}^{T^l}, R^{a,T^l}_i, P^{a,T^l}_i) + T^e \times U^a(\mathbf{z}^{T^l-1}, i)\Big] \quad (1)$$

where the objective function depends only on attack actions taken previously $\mathbf{z}^{T^l-1}$, the payoff realization $(\mathbf{R}^{a,T^l}, \mathbf{P}^{a,T^l})$ and the attack strategy $\mathbf{y}$ at current time step T^l. This objective consists of separate terms respective to each pure attack strategy $i \in \mathbf{N}$. In the objective function, $U^a_i(\mathbf{x}^{T^l}, R^{a,T^l}_i, P^{a,T^l}_i)$ is the attacker's expected utility for attacking target i with respect to $(\mathbf{x}^{T^l}, R^{a,T^l}_i, P^{a,T^l}_i)$. In addition, $U^a(\mathbf{z}^{T^l-1}, i)$ is the attacker's expected utility at each time step $t \in \{T^l+1, T^l+2, \dots, T\}$ in the execution phase with respect to the set of attacker's actions in the learning phase $(\mathbf{z}^{T^l-1}, i)$. Essentially, $U^a(\mathbf{z}^{T^l-1}, i)$ is determined as an expectation over all possible payoff realizations at each time step, as follows:

$$U^a(\mathbf{z}^{T^l-1}, i) = \int_{(\mathbf{R}^a,\mathbf{P}^a)\in\mathbf{\Omega}^{\hat{\lambda}}} \Big[f^{\hat{\lambda}}(\mathbf{R}^a, \mathbf{P}^a) \times U^a(\mathbf{x}^*(\mathbf{z}^{T^l-1}, i), \mathbf{R}^a, \mathbf{P}^a)\Big] d(\mathbf{R}^a, \mathbf{P}^a)$$

where $\mathbf{x}^*(\mathbf{z}^{T^l-1}, i)$ is the level-0 defender's optimal mixed strategy to play in the execution phase given her observation $(\mathbf{z}^{T^l-1}, i)$ in the learning phase. As explained in the previous section, this optimal defense strategy depends on whether the defender follows the MAP or Bayesian or No learning approach. In addition,

$$U^a(\mathbf{x}^*(\mathbf{z}^{T^l-1}, i), \mathbf{R}^a, \mathbf{P}^a) = \max_{j\in\mathbf{N}} U^a_j(\mathbf{x}^*(\mathbf{z}^{T^l-1}, i), R^a_j, P^a_j)$$

is the attacker's expected utility for playing a best response with respect to $(\mathbf{x}^*(\mathbf{z}^{T^l-1}, i), \mathbf{R}^a, \mathbf{P}^a)$.

Proposition 1 *At last time step T^l, given attack actions at previous time steps $\mathbf{z}^{T^l-1}$, there exists an optimal pure attack strategy solution for (1).*

As shown in (1), the optimal pure attack strategy is to attack target i with highest accumulated utility over $\{T^l, T^l+1, \dots, T\}$ for the attacker:

$$\max_{i\in\mathbf{N}} \Big[U^a_i(\mathbf{x}^{T^l}, R^{a,T^l}_i, P^{a,T^l}_i) + T^e \times U^a(\mathbf{z}^{T^l-1}, i)\Big]$$

We can discretize the payoff space $\mathbf{\Omega}^{\hat{\lambda}}$ and apply Bayesian-game algorithms to compute the attacker's utility at each time step of the execution phase, $U^a(\mathbf{z}^{T^l-1}, i)$. According to Proposition 1, we can solve (1) by iterating over all pure attack strategies to find the optimal attack one.

At time step $t < T^l$. At time step $t < T^l$, the payoff realization $(\mathbf{R}^{a,t}, \mathbf{P}^{a,t})$ is revealed to the attacker. The attacker's actions at previous time steps $\mathbf{z}^{t-1}$, which are randomly drawn from his strategies $(\mathbf{y}^1, \mathbf{y}^2, \ldots, \mathbf{y}^{t-1})$, are known to the defender. The attacker aims at finding an optimal strategy as to maximize his accumulated utility in expectation over all time steps $\{t, t+1, \ldots, T\}$, taking into account all possible future payoff realizations.

We assume that for each possible attack action at t, $i \in \mathbf{N}$, and each possible payoff realization at next time step $t+1$, $(\mathbf{R}^a, \mathbf{P}^a)$, the attacker's maximum accumulated utility in expectation over $\{t+1, t+2, \ldots, T\}$ only depends on $\{\mathbf{z}^{t-1}, i\}$ and $(\mathbf{R}^a, \mathbf{P}^a)$. We denote this optimal utility by $Q((\mathbf{z}^{t-1}, i), \mathbf{R}^a, \mathbf{P}^a)$. This assumption holds true for the last time step T^l. Then finding an optimal attack strategy at t can be represented as follows:

$$Q(\mathbf{z}^{t-1}, \mathbf{R}^{a,t}, \mathbf{P}^{a,t}) = \max_{\mathbf{y} \in \mathbf{Y}} \sum_{i \in \mathbf{N}} y_i \Big[U_i^a(\mathbf{x}^t, R_i^{a,t}, P_i^{a,t}) + \int_{(\mathbf{R}^a, \mathbf{P}^a) \in \mathbf{\Omega}^{\hat{\lambda}}} f^{\hat{\lambda}}(\mathbf{R}^a, \mathbf{P}^a) Q((\mathbf{z}^{t-1}, i), \mathbf{R}^a, \mathbf{P}^a) d(\mathbf{R}^a, \mathbf{P}^a) \Big] \tag{2}$$

of which objective function depends on $(\mathbf{z}^{t-1}, \mathbf{R}^{a,t}, \mathbf{P}^{a,t})$ and the attack strategy $\mathbf{y} \in \mathbf{Y}$ at current time step t. Here, $U_i^a(\mathbf{x}^t, R_i^{a,t}, P_i^{a,t})$ is the attacker's expected utility for attacking target $i \in \mathbf{N}$ with respect to $(\mathbf{x}^t, R_i^{a,t}, P_i^{a,t})$. This objective function consists of separate terms corresponding to each possible pure attack strategy $i \in \mathbf{N}$. Therefore, we obtain the following proposition:

Proposition 2 *At time step $t < T^l$, given attack actions at previous time steps $\mathbf{z}^{t-1}$, there exists an optimal pure attack strategy solution for (2).*

According to Proposition 2, we can approximately solve (2) by discretizing the payoff space $\mathbf{\Omega}^{\hat{\lambda}}$ of the attacker and iterating over all pure attack strategies to find an optimal one.

Limited look-ahead attack strategies

In order to compute optimal attack strategies at every time step in the learning phase, we have to take into account all possible payoff realizations in future time steps. Even we can discretize the payoff space of the attacker, there are still exponentially many possible payoff realizations over all time steps to consider. As a result, exactly computing optimal attack strategies is impractical. In this work, we propose the limited look-ahead heuristic to overcome this computational challenge. Essentially, at each time step t in the learning phase, the look-ahead heuristic only considers a small number of future time steps including t, $\{t, t+1, \ldots, t+M\}$ where $t+M < T^l$ and M is the number of time steps to look ahead. The heuristic then attempts to find an optimal attack strategy at t while assuming $t+L$ to be the last time step in the learning phase. This approach allows us to limit the number of possible future payoff realizations to consider.

Optimal defense strategy

The level-1 defender assumes the attacker is at level-1. At the end of the learning phase, the defender obtains the observation $\mathbf{z}^{T^l}$ which consists of all attack actions till time step T^l. The defender does not know the payoff realizations $\{(\mathbf{R}^{a,t}, \mathbf{P}^{a,t})\}$ where $t = 1, 2, \ldots T^l$. She is only aware of the payoff distribution f^λ over the payoff space $\mathbf{\Omega}^\lambda$ for all attacker types $\lambda \in \mathbf{\Lambda}$. Since the payoffs of the attacker at every time step are i.i.d, the defender can update the posteriori distribution over attacker types as follows:

$$\begin{aligned} p(\lambda \mid (\mathbf{x}^1, z^1), (\mathbf{x}^2, z^2), \ldots, (\mathbf{x}^{T^l}, z^{T^l})) \\ \propto p(\lambda) \times \int_{(\mathbf{R}^{a,1}, \mathbf{P}^{a,1}) \in \mathbf{\Omega}^\lambda} \Big[p(z^1 \mid \mathbf{x}^1, \mathbf{R}^{a,1}, \mathbf{P}^{a,1}, \lambda) \times \\ f^\lambda(\mathbf{R}^{a,1}, \mathbf{P}^{a,1}) \Big] d(\mathbf{R}^{a,1}, \mathbf{P}^{a,1}) \\ \times \prod_{t=2}^{T^l} \int_{(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) \in \mathbf{\Omega}^\lambda} \Big[p(z^t \mid \mathbf{x}^t, \mathbf{z}^{t-1}, \mathbf{R}^{a,t}, \mathbf{P}^{a,t}, \lambda) \times \\ f^\lambda(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) \Big] d(\mathbf{R}^{a,t}, \mathbf{P}^{a,t}) \end{aligned} \tag{3}$$

where $p(z^t \mid \mathbf{x}^t, \mathbf{z}^{t-1}, \mathbf{R}^{a,t}, \mathbf{P}^{a,t}, \lambda)$ is the probability the attacker of type λ attacks target z^t given his action history $\mathbf{z}^{t-1}$ and his payoff realization, $(\mathbf{R}^{a,t}, \mathbf{P}^{a,t})$, and the defense strategy $\mathbf{x}^t$ at t. As shown in Proposition 2, there exists an optimal pure attack strategy at each time step in the learning phase. Thus, we have: $p(z^t \mid \mathbf{x}^t, \mathbf{z}^{t-1}, \mathbf{R}^{a,t}, \mathbf{P}^{a,t}, \lambda) \in \{0, 1\}$. In addition, the defender can determine this probability by examining the optimal attack action for the level-1 attacker of type λ with respect to $(\mathbf{x}^t, \mathbf{z}^{t-1}, \mathbf{R}^{a,t}, \mathbf{P}^{a,t})$ (which is computed in the previous section) is z^t or not. Finally, we can approximately compute the posteriori distribution over attacker types in (3) by discretizing the payoff space of every attacker type $\lambda \in \mathbf{\Lambda}$.

Given the posteriori distribution of attacker types, the defender aims at computing an optimal defense strategy in the execution phase. The defender can either follow the MAP or Bayesian approach, which is similar to the level-0 defender.

Experiments: Preliminary Results

In our experiments, we aim at evaluating the solution quality of proposed strategies of players at level-0 and level-1. We run experiments on security games with the number of targets $N = 5$ and the number of defender resources $K = 3$. The number of attacker types is $|\mathbf{\Lambda}| = 3$. The probability distribution over attacker types $\mathbf{p}$ is generated by uniformly at random. For each attacker type $\lambda \in \mathbf{\Lambda}$, we consider a discrete distribution $\bar{\mathbf{f}}^\lambda$ over a discretized payoff space $\bar{\mathbf{\Omega}}^\lambda$ with $\left|\bar{\mathbf{\Omega}}^\lambda\right| = 4$. The number of time steps in the learning phase and the execution phase is $T^l = 10$ and $T^e = T - T^l = 100$. The defender payoffs and the attacker payoffs in $\bar{\mathbf{\Omega}}^\lambda$ are generated uniformly at random within the ranges $[1, 10]$ for rewards and $[-10, -1]$ for penalties at each target.

We evaluate three attack strategies: (i) **aL0-BR** (the level-0 attacker who always plays a best response); (ii) **aL1-MAP** (the level-1 attacker, assuming the defender is at level-0 and follows the MAP-based approach); and (iii) **aL1-Bayesian** (the level-1 attacker, assuming that the defender is at level-0 and follows the Bayesian approach).

In addition, we evaluate seven defense strategies. These strategies differ according to whether the defender is at

	aL0-BR	aL1-MAP	aL1-Bayesian
dL0-Prior	56.32	54.79	54.99
dL0-MAP	75.73	54.79	62.75
dL0-Bayesian	80.37	67.25	64.17
dL1-MAPvsaL1-MAP	73.79	89.74	78.28
dL1-MAPvsaL1-BAY	66.02	76.15	76.34
dL1-BAYvsaL1-MAP	68.70	78.88	70.60
dL1-BAYvsaL1-BAY	66.08	71.77	76.47

(a) Defender accumulated utility in expectation

	aL0-BR	aL1-MAP	aL1-Bayesian
dL0-Prior	-1.69	-2.34	-2.56
dL0-MAP	-10.20	-2.34	-5.97
dL0-Bayesian	-9.40	-3.67	-3.54
dL1-MAPvsaL1-MAP	-9.34	-17.66	-12.77
dL1-MAPvsaL1-BAY	-5.94	-11.70	-11.92
dL1-BAYvsaL1-MAP	-4.94	-9.42	-7.45
dL1-BAYvsaL1-BAY	-4.66	-6.08	-8.72

(b) Attacker accumulated utility in expectation

aL0-BR	aL1-Bayesian
0.68	0.32

dL0-Bayesian	dL1-MAPvsaL1-MAP
0.37	0.63

(c) Nash equilibrium

Figure 1: Game 1, Strategy evaluation

	aL0-BR	aL1-MAP	aL1-Bayesian
dL0-Prior	76.82	81.92	83.18
dL0-MAP	107.79	38.90	40.15
dL0-Bayesian	110.02	63.74	45.12
dL1-MAPvsaL1-MAP	82.06	111.15	62.43
dL1-MAPvsaL1-BAY	76.82	85.42	117.97
dL1-BAYvsaL1-MAP	82.06	98.56	64.53
dL1-BAYvsaL1-BAY	76.82	85.42	109.25

(a) Defender accumulated utility in expectation

	aL0-BR	aL1-MAP	aL1-Bayesian
dL0-Prior	130.31	125.89	121.92
dL0-MAP	112.71	205.69	201.72
dL0-Bayesian	104.62	157.85	192.15
dL1-MAPvsaL1-MAP	126.28	109.64	173.68
dL1-MAPvsaL1-BAY	130.31	123.20	102.95
dL1-BAYvsaL1-MAP	126.28	116.24	154.82
dL1-BAYvsaL1-BAY	130.31	123.20	102.14

(b) Attacker accumulated utility in expectation

aL0-BR	aL1-MAP	aL1-Bayesian
0.56	0.26	0.18

dL0-Bayesian	dL1-MAPvsaL1-MAP	dL1-MAPvsaL1-BAY
0.14	0.15	0.71

(c) Nash equilibrium

Figure 2: Game 2, Strategy evaluation

level-0 or level-1, which strategic approaches the defender follows, and what assumptions the defender has regarding the attacker. These seven defense strategies include: (i) **dL0-Prior** (the defender is at level-0 and follows the no learning approach); (ii) **dL0-MAP** (level-0 defender, MAP-based approach); (iii) **dL0-Bayesian** (level-0 defender, Bayesian approach); (iv) **dL1-MAPvsaL1-MAP** (level-1 defender, MAP-based approach, assuming the attacker plays aL1-MAP); (v) **dL1-MAPvsaL1-BAY** (level-1 defender, MAP-based approach, assuming aL1-Bayesian); (vi) **dL1-BAYvsaL1-MAP** (level-1 defender, Bayesian approach, assuming aL1-MAP); and (vii) **dL1-BAYvsaL1-BAY** (level-1 defender, Bayesian approach, assuming aL1-Bayesian).

In the following, we present results of three different games. For each game, we run 30 simulations to compute the players' accumulated utility over all time steps in expectation for playing each pair of proposed strategies. The results are shown in Figures 1, 2, and 3. In particular, Figures 1(a), 2(a), and 3(a) show the defender's accumulated utility for playing strategies (listed in the first column) against the attack strategies (listed in the first row). For example, in Figure 1(a), the defender obtains a utility of 56.32 for playing dL0-Prior against the attack strategy aL0-BR. Similarly, Figures 1(b), 2(b), and 3(b) show the attacker's accumulated utility. Finally, Figures 1(c), 2(c), and 3(c) show Nash equilibria of games constructed based on players' utility in Figures 1(a)(b), 2(a)(b), and 3(a)(b) respectively.

In Figures 1(a), 2(a), and 3(a), the defender's utility for playing either dL0-MAP or dL0-Bayesian is significantly lower when the attacker plays aL1-MAP or aL1-Bayesian than when the attacker plays aL0-BR. This result shows that when the level-0 defender follows a learning approach (i.e., MAP-based or Bayesian), she suffers a significant loss in her utility if the attacker plays deceitfully to mislead her. Meanwhile, by playing any of level-1 defense strategies (i.e., the last four defense strategies shown in the first column), the defender's utility increases significantly compared to the level-0 strategies. This result confirms the importance of taking into account deceitful attacks for the defender.

In Figures 1(b), 2(b), and 3(b), if the defender plays level-0 learning-based strategies (i.e., MAP-based or Bayesian approach), the attacker can gain a great benefit by manipulating his attacks. For example, Figure 2(b) shows that the attacker obtains an utility of 205.69 for playing the aL1-MAP strategy, which is approximately twice more than his utility for playing aL0-BR, against the dL0-MAP defense strategy. In addition, the attacker obtains a higher utility for playing any level-1 strategies than playing aL0-BR against level-0 learning-based defense strategies, i.e., dL0-MAP and dL0-Bayesian. This result implies that if the defender is at level-0 and follows MAP-based or Bayesian approach, the attacker can always gain benefit by playing deceitfully regardless of his assumption about the defender.

Finally, the Nash equilibrium result in Figures 1(c), 2(c), and 3(c) involve strategies from both level-0 and level-1 for the defender and the attacker. In addition, no pure Nash equilibrium exists in all three games.

	aL0-BR	aL1-MAP	aL1-Bayesian
dL0-Prior	203.13	202.46	204.11
dL0-MAP	276.85	233.88	210.39
dL0-Bayesian	294.80	204.56	206.20
dL1-MAPvsaL1-MAP	203.13	251.21	204.11
dL1-MAPvsaL1-BAY	220.47	220.27	262.57
dL1-BAYvsaL1-MAP	203.13	260.04	204.11
dL1-BAYvsaL1-BAY	219.23	210.84	269.91

(a) Defender accumulated utility in expectation

	aL0-BR	aL1-MAP	aL1-Bayesian
dL0-Prior	23.45	20.90	21.23
dL0-MAP	7.35	30.12	23.07
dL0-Bayesian	2.18	21.52	21.85
dL1-MAPvsaL1-MAP	23.45	10.80	21.23
dL1-MAPvsaL1-BAY	19.66	26.12	9.54
dL1-BAYvsaL1-MAP	23.45	7.36	21.23
dL1-BAYvsaL1-BAY	19.52	23.36	5.84

(b) Attacker accumulated utility in expectation

aL0-BR	aL1-MAP	aL1-Bayesian
0.16	0.53	0.31

dL0-MAP	dL1-MAPvsaL1-BAY	dL1-BAYvsaL1-MAP
0.25	0.28	0.47

(c) Nash equilibrium

Figure 3: Game 3, Strategy evaluation

Summary

In this work, we studied the problem of deceitful attacks in security games. We proposed a new repeated security game model in which the defender can collect attack data over multiple time steps to learn the attacker type and then plan patrolling strategies accordingly. Meanwhile, the attacker can manipulate his attacks to deceit the defender about his type, leading her to choose patrolling strategies which would benefit the attacker the most. We then propose new game-theoretic algorithms to compute optimal strategies for both players, taking into account all future possibilities given the attacker's payoffs dynamically change over time. We provide preliminary experimental results, showing that our defense solutions can effectively address deceitful attacks.

Acknowledgements

This work was supported in part by MURI grant W911NF-13-1-0421 from the US Army Research Office.

References

Balcan, M.-F.; Blum, A.; Haghtalab, N.; and Procaccia, A. D. 2015. Commitment without regrets: Online learning in stackelberg security games. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, 61–78. ACM.

Barreno, M.; Nelson, B.; Sears, R.; Joseph, A. D.; and Tygar, J. D. 2006. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, 16–25. ACM.

Barreno, M.; Nelson, B.; Joseph, A. D.; and Tygar, J. 2010. The security of machine learning. *Machine Learning* 81(2):121–148.

Basilico, N.; Gatti, N.; and Amigoni, F. 2009. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, 57–64. International Foundation for Autonomous Agents and Multiagent Systems.

Bier, V. M., and Zhuang, J. 2008. Secrecy and deception in anti-terrorism resource allocation and policy implication.

Blum, A.; Haghtalab, N.; and Procaccia, A. D. 2014. Learning optimal commitment to overcome insecurity. In *Advances in Neural Information Processing Systems*, 1826–1834.

Brown, G.; Carlyle, M.; Diehl, D.; Kline, J.; and Wood, K. 2005. A two-sided optimization for theater ballistic missile defense. *Operations research* 53(5):745–763.

Brückner, M., and Scheffer, T. 2011. Stackelberg games for adversarial prediction problems. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 547–555. ACM.

Brückner, M.; Kanzow, C.; and Scheffer, T. 2012. Static prediction games for adversarial learning problems. *Journal of Machine Learning Research* 13(Sep):2617–2654.

Fang, F.; Nguyen, T. H.; Pickles, R.; Lam, W. Y.; Clements, G. R.; An, B.; Singh, A.; Tambe, M.; and Lemieux, A. 2016. Deploying paws: Field optimization of the protection assistant for wildlife security. In *AAAI*, 3966–3973.

Farrell, J., and Rabin, M. 1996. Cheap talk. *The Journal of Economic Perspectives* 10(3):103–118.

Gholami, S.; Ford, B.; Fang, F.; Plumptre, A.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Nsubaga, M.; and Mabonga, J. 2017. Taking it for a test drive: a hybrid spatio-temporal model for wildlife poaching prediction evaluated through a controlled field test. In *Proceedings of the European Conference on Machine Learning & Principles and Practice of Knowledge Discovery in Databases, ECML PKDD*.

Guo, Q.; An, B.; Bosansky, B.; and Kiekintveld, C. 2017. Comparing strategic secrecy and stackelberg commitment in security games. IJCAI.

Haghtalab, N.; Fang, F.; Nguyen, T. H.; Sinha, A.; Procaccia, A. D.; and Tambe, M. 2016. Three strategies to success: Learning adversary models in security games. In *IJCAI*, 308–314.

Hendricks, K., and McAfee, R. P. 2006. Feints. *Journal of Economics & Management Strategy* 15(2):431–456.

Kar, D.; Ford, B.; Gholami, S.; Fang, F.; Plumptre, A.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Nsubaga, M.; et al. 2017. Cloudy with a chance of poaching: Adversary behavior modeling and forecasting with real-world poaching data. In *Proceedings of the 16th Conference on*

Autonomous Agents and MultiAgent Systems, 159–167. International Foundation for Autonomous Agents and Multiagent Systems.

Letchford, J., and Vorobeychik, Y. 2011. Computing randomized security strategies in networked domains. *Applied Adversarial Reasoning and Risk Modeling* 11:06.

Lowd, D., and Meek, C. 2005. Adversarial learning. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, 641–647. ACM.

Nguyen, T. H.; Sinha, A.; Gholami, S.; Plumptre, A.; Joppa, L.; Tambe, M.; Driciru, M.; Wanyama, F.; Rwetsiba, A.; Critchlow, R.; et al. 2016. Capture: A new predictive anti-poaching tool for wildlife protection. In *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, 767–775. International Foundation for Autonomous Agents and Multiagent Systems.

Rabinovich, Z.; Jiang, A. X.; Jain, M.; and Xu, H. 2015. Information disclosure as a means to security. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, 645–653. International Foundation for Autonomous Agents and Multiagent Systems.

Tambe, M. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.

Xu, H.; Rabinovich, Z.; Dughmi, S.; and Tambe, M. 2015. Exploring information asymmetry in two-stage security games. In *AAAI*, 1057–1063.