24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Tackling Imitative Attacker Deception in Repeated Bayesian Stackelberg Security Games

Thanh H. Nguyen[1] and **Andrew Butler**[2] and **Haifeng Xu**[3]

Abstract. Uncertainty is one of the key challenges in Stackelberg security games (SSGs) — a well known class of games which have been used to solve various real-world problems in security domains such as airport security and wildlife protection. In these domains, security agencies (defender) are generally uncertain about underlying characteristics of attackers such as the attackers' preference or behavior. Previous work in SSGs has proposed different learning methods to reduce uncertainty about attackers based on historical attack data. However, since these learning algorithms depend on the attack data, a clever attacker can manipulate its attacks to influence the learning outcome. Such deception of the attacker could lead to ineffective defense strategies that favor the attacker.

This work studies the strategic deception of the attacker with private type information in a repeated Bayesian Stackelberg security game (RBSSG) setting. We investigate a basic deception strategy, named *imitative attacker deception* — that is the attacker pretends to have a different type and consistently plays according to that deceptive type throughout the whole time horizon. We provide four main contributions. First, we present a detailed equilibrium computation and analysis of standard RBSSGs with a *non-deceptive* attacker. To our knowledge, our work is the first to present an exact algorithm to compute an RBSSG equilibrium. Second, we introduce a new counter-deception algorithm, built on our equilibrium computation of standard RBSSGs, to address the attacker's deception. Third, we introduce two new heuristics to overcome the computational challenge of the exact algorithm: one heuristic limits the number of time steps to look ahead and the other limits the number of observational attack histories in consideration. Fourth, we conduct extensive experiments, showing the significant loss of the defender and benefit of the attacker as a result of the attacker deception. We also show that our counter-deception algorithm can help in substantially diminishing the impact of the attacker's deception on the players' utility.

1 Introduction

Stackelberg security games (SSGs) have been widely applied for solving many real-world security problems. There have been several deployed applications of SSGs in security domains such as ferry protection and wildlife conservation [4, 22]. One of the key characteristics in these domains is the defender's uncertainty about the attackers' preferences and behaviors. Through repeated interactions with the attacker, the defender can collect observational attack data to gain knowledge about these characteristics of the attacker. Such knowledge is then incorporated into generating an effective defense strategy [1, 3, 8, 10, 11, 13, 15, 16, 19, 24]. In wildlife security, for example, rangers have to conduct patrols within a conservation area to protect animals from poachers. During patrols, rangers can collect poaching signs such as snares and use the collected poaching data to build a behavior model of the poachers. The rangers then design patrols, assuming the poachers follow the learnt behavior model [4, 16].

However, since the defender relies on the historical attack data, a clever attacker can manipulate its attacks to *fool* the defender's learning algorithms. That is, the attacker would choose actions that do not result in the highest immediate reward, but influence the learning outcome to a long-term benefit for the attacker. The deceptive attacker aims at finding an optimal strategic deception that maximizes its total accumulated expected reward over time. Such deception of the attacker could deteriorate the effectiveness of the defender's strategy. Note that, in standard SSGs, the attacker is assumed to be *non-manipulative* in the sense that it only attempts to maximize its *immediate* reward, regardless of the defender's learning.

Motivated by this deception challenge, this work studies a basic deception strategy of the attacker, named *imitative attacker deception* in RBSSGs. In RBSSGs, the defender is uncertain about the attacker type. However, he knows a prior distribution over the attacker types. At each time step, the defender commits to a mixed strategy and the attacker responds accordingly. The attacker responses at previous time steps are then used to update the defender's belief about the attacker types at current step. The defender then determines a new defense strategy based on his updated belief. To influence the defender's learning outcome towards the attacker's benefit, the attacker can imitate a type (which may be different from its true type) and consistently plays the game according to that type.

In this work, we provide four main contributions. First, we propose a new game-theoretic algorithm to compute an exact equilibrium of the standard RBSSGs in which the attacker is non-manipulative. This computation equilibrium serves as a basis for us to develop new counter-deception algorithms to address the attacker's deception. To our knowledge, this is the first exact algorithm for computing an RBSSG equilibrium. Our algorithm leverages the linearity property of the players' utility to provide a Mixed Integer Linear Program (MILP) to compute an equilibrium of the game.

Second, to address the attacker imitative deception, we introduce a new game-theoretic algorithm, which is an extension of the MILP formulation for standard RBSSG equilibrium, to find an optimal counter-deception strategic plan for the defender. In our algorithm, the defender chooses a defense plan designed based on his observations about the attacker's historical attack activities. Third, we introduce two new heuristics: *limited-look-ahead* and *limited-memory-length* heuristics. The main ideas are to only consider a small number of future time steps or a small number of observation histories when

[1] University of Oregon, USA, email: tnguye11@uoregon.edu
[2] University of Oregon, USA, email: abutler9@uoregon.edu
[3] University of Virginia, USA, email: hx4ad@virginia.edu

computing the defender's strategic plan. Therefore, our heuristics can overcome the challenges of an exponential number of observation histories and future possibilities in the proposed exact algorithm.

Fourth, we provide a detailed analysis on both the runtime performance and solution quality of our proposed algorithms. We show that the attacker's imitative deception could cause a significant loss in utility for the defender if the defender does not address the attacker deception. Our results also show that our counter-deception game-theoretic algorithm helps in reducing such loss drastically.

2 Related Work

Most relevant to our work is the extensive recent study on deception (on the attacker side) in SSGs [6, 17, 18] or more generally Stackelberg games [7]. In [6, 7, 17], in particular, they also focused on the follower's imitative deception. However, the existing work all considered *single-shot* Stackelberg games whereas we study the attacker deception in *repeated* Bayesian Stackelberg games. The work in [18] considered repeated security games but in a *simultaneous* game setting. This existing work then focused on computing and analyzing the Bayesian Nash equilibrium of the game. Our work, on the other hand, studies the attacker deception in a *Stackelberg* (sequential-move) setting, which has completely different characteristics.

Our work contributes to the rich literature on understanding deception in security domains [5, 21]. In particular, a line of recent work examines deception from the *defender's side* and study how to mislead the attacker's decision by exploiting the defender's knowledge regarding uncertainties [9, 20, 23, 25]. However, our focus here is to study the natural counterpart, i.e., *attacker deception*. Our work also relates to the *causative-attack* problem in adversarial learning in which a data generator intentionally transforms the training data to trick machine learning algorithms of a learner [2, 14]. While existing work in adversarial learning mainly focuses on prediction accuracy, both the defender and attacker in our deception problem face a *decision making* problem of optimizing their own utilities and the defender has the learn to make his optimal decisions.

On the technical side, our first technical result presents an algorithm for solving a repeated Bayesian Stackelberg security game (RBSSG) based on optimization programs. In, [15], they considered a similar problem but used the Monte Carlo tree search to design a heuristic, and possibly sub-optimal, algorithm whereas our algorithm is exact. To our knowledge, this is the first exact algorithm for solving RBSSGs. Our work is also related to the large literature for solving extensive games since repeated Stackelberg games can be viewed as a special class of extensive-form games. Indeed, our accelerated heuristic algorithm, *limited-look-ahead*, are inspired by the limited look ahead idea used in extensive form games (see, e.g., [12]).

3 Repeated Bayesian Stackelberg Security Games

We first provide preliminaries of standard repeated Bayesian Stackelberg security games (RBSSGs) in which the attacker is non-deceptive — the attacker plays a myopic best response at each time step. We then present our new algorithm to compute an exact RBSSG equilibrium. Table 1 summarizes notations used throughout our paper.

3.1 Preliminaries

In Stackelberg security games (SSGs), there is a set of N important targets $[N] = \{1, 2, \ldots, N\}$. A defender has to assign a limited number of security resources to the targets, denoted by K with

Notation	Description
N	Number of targets
T	Number of time steps
K	Number of security resources
$\{1, 2, \ldots, \Lambda\}$	Set of attacker types
R^d_i, P^d_i	Defender's reward and penalty at target i
R^λ_i, P^λ_i	Attacker type λ's reward and penalty at i
p_λ	Prior probability of attacker type λ
$\mathbf{x} = \{x_i\}_{i=1}^N$	x_i is the defender's coverage probability at i
$\mathbf{h}_t$	Attack history at time step t: $(i_1, \ldots, i_{t-1})$
$\mathbf{h}_{t',\ldots,t}$	Partial attack history $(i_{t'}, \ldots, i_{t-1})$
$\pi = \{\mathbf{x}(\mathbf{h}_t)\}$	Defender's strategy plan: $\mathbf{x}(\mathbf{h}_t)$ is defender's mixed strategy w.r.t $\mathbf{h}_t$, for all $\mathbf{h}_t$
$\mathbf{as} = \{\mathbf{as}^\lambda\}_\lambda$	$\mathbf{as}^\lambda = \{as^\lambda_t\}_{t=1}^T$ is an attack sequence of type λ; $as^\lambda_t = i^\lambda_t$ is the attacked target at t
$\mathbf{as}^\lambda_{1\ldots t}$	An attack sequence of type λ up to step t
$\Lambda(\mathbf{h}_t)$	Set of attacker types λ s.t. $\mathbf{as}^\lambda_{1\ldots t-1} \equiv \mathbf{h}_t$.
$\beta(\lambda \mid \mathbf{h}_t)$	Defender's updated belief on attacker types
$EU^d(\mathbf{x}, i)$	Defender's immediate expected utility at i
$EU^\lambda(\mathbf{x}, i)$	Type λ's immediate expected utility at i
$EU^{d,t}_\lambda$	Abbreviation of $EU^d(\mathbf{x}(\mathbf{h}_t), i^\lambda_t)$
$U^d_t(\mathbf{h}_t)$	Abbreviation of defender's total expected utility: $U^d_t(\mathbf{x}(\mathbf{h}_t), \{i^\lambda_t\}_\lambda, \mathbf{h}_t)$ w.r.t attack history $\mathbf{h}_t$

Table 1: Summary of important notations

$K < N$, to protect these targets. A pure strategy of the defender is an allocation of the security resources to the targets. A mixed strategy of the defender is a probabilistic distribution over all of his pure strategies. In this work, we focus on the scenario in which there is no scheduling constraint on the defender's strategy. In this case, a mixed strategy of the defender can be equivalently represented as a marginal coverage probability vector, $\mathbf{x} = \{x_1, x_2, \ldots, x_N\}$, where $x_i \in [0, 1]$ is the probability the defender protects target i and $\sum_i x_i = K$. In SSGs, an attacker, who is aware of $\mathbf{x}$, responds by attacking a target with the highest expected utility for the attacker.

In the Bayesian setting, there is a set of the attacker types: $\{1, 2, \ldots, \Lambda\}$. Each type λ is associated with a prior probability $p_\lambda \in [0, 1]$ where $\sum_\lambda p_\lambda = 1$. At the beginning, a type of the attacker is randomly drawn from this prior distribution to play the game. The attacker knows its true type. On the other hand, the defender does not know which attacker type is playing. However, the defender knows the prior distribution over the attacker types $\{p_\lambda\}$. When the attacker of type λ attacks target i, if the defender is protecting that target, then the attacker receives a penalty of P^λ_i while the defender gets a reward of R^d_i. Conversely, if the defender is not protecting i, then the attacker obtains a reward of $R^\lambda_i > P^\lambda_i$ and the defender receives a penalty of $P^d_i < R^d_i$. Given a mixed strategy $\mathbf{x}$ of the defender, when the attacker attacks target i, the attacker (of type λ) and defender's expected utility can be formulated as follows:

$$EU^d(\mathbf{x}, i) = x_i R^d_i + (1 - x_i) P^d_i$$
$$EU^\lambda(\mathbf{x}, i) = x_i P^\lambda_i + (1 - x_i) R^\lambda_i$$

In the repeated setting, the game has T time steps. At each step t, the defender chooses a mixed strategy to play and the attacker chooses to attack a target in response. We denote by $\mathbf{h}_t = \{i_1, i_2, \ldots, i_{t-1}\}$ the attack history at step t with $i_{t'}$ is the attacked target at time step $t' \leq t - 1$. A *strategy plan* of the defender consists of all of his mixed strategies $\pi = \{\mathbf{x}(\mathbf{h}_t)\}$ with respect to all attack histories $\{\mathbf{h}_t\}$. An *attack sequence* of the attacker of type λ is denoted by $\mathbf{as}^\lambda = \{i_1^\lambda, i_2^\lambda, \ldots, i_T^\lambda\}$ where i_t^λ is the attacked target at step t. Given $(\pi, \{\mathbf{as}^\lambda\}_\lambda)$, the defender's total expected utility can be computed using backward induction as follows:

At time step T, given an attack history $\mathbf{h}_T$, the defender's total expected utility is computed as follows:

$$U_T^d(\mathbf{x}(\mathbf{h}_T), \{i_T^\lambda\}_\lambda, \mathbf{h}_T) = \sum\nolimits_\lambda \beta(\lambda \mid \mathbf{h}_T) EU^d(\mathbf{x}(\mathbf{h}_T), i_T^\lambda)$$

which is an expectation over all attacker types. The defender's updated belief, $\{\beta(\lambda \mid \mathbf{h}_T)\}_{\lambda=1}^\Lambda$, which is a posterior distribution over attacker types given the observation history $\mathbf{h}_T$, is computed using the Bayes approach as follows:

$$\beta(\lambda \mid \mathbf{h}_T) \propto p_\lambda \text{ if } \mathbf{h}_T \equiv \{i_1^\lambda, i_2^\lambda, \ldots, i_T^\lambda\}$$
$$\beta(\lambda \mid \mathbf{h}_T) = 0, \text{ otherwise.}$$

At time step $t < T$, given an attack history $\mathbf{h}_t$, the defender's total expected utility is computed as follows:

$$\begin{aligned} U_t^d(\mathbf{x}(\mathbf{h}_t), \{i_t^\lambda\}_\lambda, \mathbf{h}_t) &= \sum\nolimits_\lambda \beta(\lambda \mid \mathbf{h}_t) EU^d(\mathbf{x}(\mathbf{h}_t), i_t^\lambda) \\ &+ \sum\nolimits_\lambda \beta(\lambda \mid \mathbf{h}_t) U_{t+1}^d(\mathbf{x}(\mathbf{h}_t, i_t^\lambda), \{i_{t+1}^\lambda\}_\lambda, (\mathbf{h}_t, i_t^\lambda)) \end{aligned}$$

which is the sum of two terms: (i) the first term is the immediate expected utility as a result of the pair $(\mathbf{x}(\mathbf{h}_t), \{i_t^\lambda\}_\lambda)$ at current step t; and (ii) the second term is the future expected utility. The updated belief $\beta(\lambda \mid \mathbf{h}_t)$ is computed similar to $\beta(\lambda \mid \mathbf{h}_T)$.

A key assumption of RBSSGs is that the attacker is *myopic*. That is, the attacker attacks a target at which the attacker's immediate expected utility is highest. Based on this assumption and the computation of the defender's total expected utility, we can define the repeated Bayesian Stackelberg equilibrium of the game.

Definition 1 (Repeated Bayesian Stackelberg Equilibrium)**.** *A pair of strategies, $(\pi^*, \mathbf{as}^*(\pi^*))$, forms an equilibrium if and only if:*

- *Attacker plays a myopic best response: for any $\mathbf{x}$,*

$$as_t^{\lambda,*}(\mathbf{x}) \in \operatorname*{argmax}_i EU^\lambda(\mathbf{x}, i), \forall \lambda$$

- *Defender commits to the best strategy plan:*

$$\mathbf{x}^*(\mathbf{h}_t) \in \operatorname*{argmax}_{\mathbf{x}} U_t^d(\mathbf{x}, \{as_t^{\lambda,*}(\mathbf{x})\}_\lambda, \mathbf{h}_t), \forall \textit{ feasible } \mathbf{h}_t$$

An attack history $\mathbf{h}_t = \{i_1, i_2, \ldots, i_{t-1}\}$ is feasible if it matches an attacker type's sequence of myopic best responses up to $t-1$. That is, there is a type λ such that $i_{t'} \in \operatorname{argmax}_i EU^\lambda(\mathbf{x}(\mathbf{h}_{t'}), i)$ where the history $\mathbf{h}_{t'} = \{i_1, i_2, \ldots, i_{t'-1}\}$, $\forall t' \leq t$.

3.2 Equilibrium Computation

According to Definition 1, finding an equilibrium of the game involves multiple nested optimization problems, which are not easy to solve. In this section, we show that these nested optimization problems can be merged into one single mixed integer linear program. We first present Lemma 1, showing that given a fixed updated belief, the defender's total expected utility at each time step is a linear combination of multiple separate terms. Each term refers to the defender's immediate expected utility with respect to the defender's mixed strategy and the attacker's attack choice at that time step.

Lemma 1. *Let's suppose the defender and attacker play $(\pi, \mathbf{as} = \{\mathbf{as}^\lambda\}_\lambda)$. At each step t, given a feasible attack history $\mathbf{h}_t$, the defender's total expected utility can be represented as follows:*

$$U_t^d(\mathbf{h}_t) = \sum_\lambda \beta(\lambda \mid \mathbf{h}_t) \sum_{t'=t}^T EU_\lambda^{d,t'} \tag{1}$$

where $U_t^d(\mathbf{h}_t)$ is the abbreviation of the total expected utility $U_t^d(\mathbf{x}(\mathbf{h}_t), \{i_t^\lambda\}_\lambda, \mathbf{h}_t)$ and $EU_\lambda^{d,t'}$ is the abbreviation of the immediate expected utility $EU^d(\mathbf{x}(\mathbf{h}_{t'}), i_{t'}^\lambda)$ with respect to the type λ.[4] *The attack history $\mathbf{h}_t$ is feasible if $\mathbf{h}_t \equiv \{i_1^\lambda, \ldots, i_{t-1}^\lambda\}$ for some λ.*

Proof. We prove the statement using backward induction. We can easily show that Equation (1) is correct when $t = T$. Suppose that (1) holds true for some $t + 1 \leq T$, we will show that it is true for $t' = t$.

At time step t, given a feasible history $\mathbf{h}_t$, we denote by $\Lambda(\mathbf{h}_t) = \{\lambda : \mathbf{h}_t \equiv (i_1^\lambda, \ldots, i_{t-1}^\lambda)\}$ the set of attacker types which have the sequence of attacks before t identical to $\mathbf{h}_t$. We also denote by $Z(\mathbf{h}_t) = \{i_t^\lambda : \lambda \in \Lambda(\mathbf{h}_t)\}$ the set of targets which are chosen to attack at time step t by the attacker types in $\Lambda(\mathbf{h}_t)$. The defender's total expected utility is computed as follows:

$$U_t^d(\mathbf{h}_t) = \sum_{\lambda \in \Lambda(\mathbf{h}_t)} \beta(\lambda \mid \mathbf{h}_t) EU_\lambda^{d,t} + \sum_{\lambda \in \Lambda(\mathbf{h}_t)} \beta(\lambda \mid \mathbf{h}_t) U_{t+1}^d(\mathbf{h}_t, i_t^\lambda)$$

$$= \underbrace{\sum_{\lambda \in \Lambda(\mathbf{h}_t)} \beta(\lambda \mid \mathbf{h}_t) EU_\lambda^{d,t}}_{\text{first term}} + \underbrace{\sum_{i \in Z(\mathbf{h}_t)} \left[\sum_{\lambda \in \Lambda(\mathbf{h}_t, i)} \beta(\lambda \mid \mathbf{h}_t) \right] U_{t+1}^d(\mathbf{h}_t, i)}_{\text{second term}}$$

in which the first term accounts for the immediate expected utility and the second term accounts for the future expected utility. Since (1) is true for $t + 1$, this second term can be reformulated as:

$$\sum_{i \in Z(\mathbf{h}_t)} \left[\sum_{\lambda \in \Lambda(\mathbf{h}_t, i)} \beta(\lambda \mid \mathbf{h}_t) \right] \left[\sum_{\lambda \in \Lambda(\mathbf{h}_t, i)} \beta(\lambda \mid \mathbf{h}_t, i) \sum_{t'=t+1}^T EU_\lambda^{d,t'} \right]$$

Note that, the defender's belief at each step can be updated using the Bayesian approach, as shown below:

$$\beta(\lambda \mid \mathbf{h}_t) = \frac{p_\lambda}{\sum\limits_{\lambda' \in \Lambda(\mathbf{h}_t)} p_{\lambda'}}, \forall \lambda \in \Lambda(\mathbf{h}_t)$$

$$\beta(\lambda \mid \mathbf{h}_t, i) = \frac{p_\lambda}{\sum\limits_{\lambda' \in \Lambda(\mathbf{h}_t, i)} p_{\lambda'}}, \forall \lambda \in \Lambda(\mathbf{h}_t, i)$$

As a result, the second term can be reformulated as follows:

$$\sum_{i \in Z(\mathbf{h}_t)} \left[\frac{\cancel{\sum\limits_{\lambda' \in \Lambda(\mathbf{h}_t, i)} p_{\lambda'}}}{\sum\limits_{\lambda' \in \Lambda(\mathbf{h}_t)} p_{\lambda'}} \right] \left[\sum_{\lambda \in \Lambda(\mathbf{h}_t, i)} \frac{p_\lambda}{\cancel{\sum\limits_{\lambda' \in \Lambda(\mathbf{h}_t, i)} p_{\lambda'}}} \sum_{t'=t+1}^T EU_\lambda^{d,t'} \right]$$

$$= \sum_{i \in Z(\mathbf{h}_t)} \sum_{\lambda \in \Lambda(\mathbf{h}_t, i)} \frac{p_\lambda}{\sum\limits_{\lambda' \in \Lambda(\mathbf{h}_t)} p_{\lambda'}} \sum_{t'=t+1}^T U_\lambda^{d,t'}$$

$$= \sum_{\lambda \in \Lambda(\mathbf{h}_t)} \beta(\lambda \mid \mathbf{h}_t) \sum_{t'=t+1}^T U_\lambda^{d,t'}$$

Finally, by combining the first term and the reformulation of the second term, we obtain Equation (1) for time step t. $\square$

[4] We use these abbreviations when the context is clear for simplification.

Based on Lemma 1, we introduce the MILP (2–9) which aims at maximizing the defender's total expected utility at time step $t = 1$. As we prove later, (2–9) returns an equilibrium of the game. Essentially, this MILP has the following variables:

- $EU^{d,t}_{\lambda}$: represents the defender's immediate expected reward at step t with respect to the attacker type λ.
- HU^{λ}_{t}: is the attacker's immediate highest expected utility at t.
- $z^{\lambda}_{t}(i)$: the binary variable which represents if the attacker type λ attacks target i at step t (i.e., $z^{\lambda}_{t}(i) = 1$) or not (i.e., $z^{\lambda}_{t}(i) = 0$).
- $\{x^{\lambda}_{t}(i)\}_i$: represents the defender strategy that type λ faces at t.
- $\{x_i(\mathbf{h}_t)\}$: refer to the defender's strategy plan.

$$\max \sum\nolimits_{\lambda} p_{\lambda} \sum\nolimits_{t=1}^{T} EU^{d,t}_{\lambda} \qquad (2)$$

$$\text{s.t. } HU^{\lambda}_{t} \geq EU^{\lambda}(\mathbf{x}^{\lambda}_{t}, i), \qquad \forall \lambda, t, i \quad (3)$$

$$HU^{\lambda}_{t} \leq EU^{\lambda}(\mathbf{x}^{\lambda}_{t}, i) + (1 - z^{\lambda}_{t}(i))M, \qquad \forall \lambda, t, i \quad (4)$$

$$EU^{d,t}_{\lambda} \leq EU^{d}(\mathbf{x}^{\lambda}_{t}, i) + (1 - z^{\lambda}_{t}(i))M, \qquad \forall i \quad (5)$$

$$\forall \mathbf{h}_t = (i_1, i_2, \ldots, i_{t-1}) \text{ and } \forall i \in [N]:$$

$$x^{\lambda}_{t}(i) \leq x_i(\mathbf{h}_t) + (t - 1 - \sum\nolimits_{t'<t} z^{\lambda}_{t'}(i_{t'}))M \qquad (6)$$

$$x^{\lambda}_{t}(i) \geq x_i(\mathbf{h}_t) - (t - 1 - \sum\nolimits_{t'<t} z^{\lambda}_{t'}(i_{t'}))M \qquad (7)$$

$$\sum\nolimits_{i} z^{\lambda}_{t}(i) = 1, z^{\lambda}_{t}(i) \in \{0, 1\}, \qquad \forall \lambda, t, i \quad (8)$$

$$\sum\nolimits_{i} x_i(\mathbf{h}_t) \leq K, x_i(\mathbf{h}_t) \in [0, 1], \qquad \forall (\mathbf{h}_t, i) \quad (9)$$

Constraints (3–4) guarantee that HU^{λ}_{t} is the immediate highest expected utility for the attacker type λ and the attacker attacks target i at step t (i.e., $z^{\lambda}_{t}(i) = 1$) only if i is its myopic best response. Constraint (4) indicates that the immediate expected utility of the defender, $EU^{d,t}_{\lambda}$, is equal to the defender's expected utility at target i if the attacker type λ attacks this target at step t. Constraint (6–7) ensures that the defender strategy which the type λ faces at t is the same as the defender strategy with respect to the attack history $\mathbf{h}_t$ if the attacker's sequence of attacks matches $\mathbf{h}_t$ (i.e., $\lambda \in \Lambda(\mathbf{h}_t)$ or $z^{\lambda}_{t'}(i_{t'}) = 1, \forall t' < t$). Constraints (8–9) represent feasibility conditions on the players' strategies. Finally, M is a very large constant.

Theorem 1. *The* MILP *(2–9) returns an equilibrium of the game.*

Proof. We prove this statement using forward induction. It is obvious that (2–9) returns the best strategy of the defender at time step $t = 1$ since (2–9) aims at maximizing the defender's total expected utility at $t = 1$. Suppose that this best-strategy statement is true with all $t' \leq t$ for some t, we show that it holds true for $t + 1$. Given a fixed partial strategy plan $\pi_{1\ldots t} = \{\mathbf{x}^*(\mathbf{h}_{t'})\}$ where $t' \leq t$, the partial sequence of best-response attacks for each attacker type λ, $\mathbf{as}^{\lambda,*}_{1\ldots t} = \{i^{\lambda,*}_{1}, \ldots, i^{\lambda,*}_{t}\}$, can be predetermined. We denote by $\mathbf{H}_{t+1} = \{\mathbf{h}_{t+1} : \mathbf{h}_{t+1} \equiv \mathbf{as}^{\lambda,*}_{1\ldots t}, \text{ for some } \lambda\}$ the set of all myopic-best-response histories of all attacker types prior to time step $t + 1$.

As a result, the objective of the MILP (2–9) consists of two terms: (i) the first term $\sum_{\lambda} p_{\lambda} \sum_{t'=1}^{t} U^{d,t'}_{\lambda}$ is fixed; and (ii) the second term $\sum_{\lambda} p_{\lambda} \sum_{t'=t+1}^{T} U^{d,t'}_{\lambda}$ can be decomposed as follows:

$$\sum_{\lambda} p_{\lambda} \sum_{t'=t+1}^{T} U^{d,t'}_{\lambda} = \sum_{\mathbf{h}_{t+1} \in \mathbf{H}_{t+1}} \sum_{\lambda \in \Lambda(\mathbf{h}_{t+1})} p_{\lambda} \sum_{t'=t+1}^{T} U^{d,t'}_{\lambda}$$

$$= \sum_{\mathbf{h}_{t+1} \in \mathbf{H}_{t+1}} \left[\sum_{\lambda' \in \Lambda(\mathbf{h}_{t+1})} p_{\lambda'} \right] \left[\sum_{\lambda \in \Lambda(\mathbf{h}_{t+1})} \beta(\lambda \mid \mathbf{h}_{t+1}) \sum_{t'=t+1}^{T} U^{d,t'}_{\lambda} \right]$$

in which each term only depends on the history $\mathbf{h}_{t+1}$. Therefore, the MILP (2–9) can be decomposed into multiple following MILPs, each maximizes the defender's total expected utility at $\mathbf{h}_{t+1}$.

$$\max \sum_{\lambda \in \Lambda(\mathbf{h}_{t+1})} \beta(\lambda \mid \mathbf{h}_{t+1}) \sum_{t'=t+1}^{T} U^{d,t'}_{\lambda}$$

$$\text{s.t. constraints (3-9) are satisfied w.r.t } t' \geq t + 1.$$

Each of these MILPs returns a best strategy of the defender with respect to an attack history $\mathbf{h}_{t+1} \in \mathbf{H}_{t+1}$. □

4 Attacker Imitative Deception and An Example

Since the defender's equlibrium strategy plan is designed for handling a myopic (non-manipulative) attacker, this plan may be vulnerable against a manipulative attacker. That is, the attacker can play deceptively to mislead the defender about the attacker's true type. In this work, we focus on a basic imitative deception, i.e., the attacker pretends to have a different type and plays consistently according to that type in the whole time horizon. The attacker's goal is to lead the defender to learning a wrong posterior distribution over attacker types, which eventually benefits the attacker. In particular, given the defender's equilibrium strategy plan, π^*, the total expected utility of the attacker type λ for mimicking a type λ' is computed as follows:

$$\sum\nolimits_{t=1}^{T} EU^{\lambda}(\mathbf{x}^*(\mathbf{h}^{\lambda'}_{t}), i^{\lambda'}_{t})$$

where $i^{\lambda'}_{t}$ is the best response of the attacker type λ' at step t and $\mathbf{h}^{\lambda'}_{t} \equiv (i^{\lambda'}_{1}, \ldots, i^{\lambda'}_{t-1})$ is the sequence of best responses of attacker type λ' upto step $t - 1$. The attacker type λ will iteratively search through the set of the attacker types to find a deceptive type λ' to imitate so as to maximize its total expected utility. Example 1 shows that the attacker obtains a significant benefit while the defender suffers a significant loss due to the attacker's deception.

Example 1. *Let's consider a simple 2-target games with two attacker types and two time steps. The prior distribution over the attacker types is* $\{0.5, 0.5\}$. *The game payoff is shown in Table 2. Each*

	Target 1	Target 2
Rewards	2, 2, 3	6, 8, 9
Penalties	-8, -2, -4	-10, -8, -1

Table 2: An example of a 2-target game.

cell represents a payoff tuple of the defender, the attacker type 1*, and type* 2*, respectively. For example, at target* 1*, the reward and penalty of the defender are* 2 *and* -8*, respectively. The reward and penalty of attacker type 1 and 2 are* $(2, -2)$ *and* $(3, -4)$*, respectively. The equilibrium strategies of the defender are determined as follows:*

1. *At step* 1*, the defender plays the mixed strategy* $(.2353, .7647)$.
2. *If the defender observes the attacker attacks target* 1 *at step* 1*, then the defender plays* $(.5, .5)$ *at step* 2.
3. *If the defender observes the attacker attacks target* 2 *at step* 1*, then the defender plays* $(.2353, .7647)$ *at step* 2.

Non-deceptive attacker. *If the attacker plays non-deceptively, the attacker type 1 will attack target* 1 *since its expected utility at target* 1 *is* $2 \times (1 - 0.2353) + (-2) \times 0.2353 = 1.0588$ *which is greater than that at target* 2*, which is* $8 \times (1 - 0.7647) + (-8) \times 0.7647 = -4.2352$*. On the other hand, the attacker type 2 will attack target* 2*.*

As a result, at the end of time step 1*, the defender can infer that type* 1 *is playing if he observes target* 1 *is attacked. Otherwise, if he observes target* 2 *is attacked, the defender knows type* 2 *is playing. Thus, the defender can choose an optimal defense strategy at step* 2 *according to his updated belief, which is* $(.5, .5)$ *if he observes that target* 1 *is attacked and* $(.2353, .7647)$*, otherwise.*

Consequently, the defender obtains a total expected utility of -1.5883 *while the attacker type* 1 *and type* 2 *receive a total expected utility of* 1.0588 *and* 2.706*, respectively. The attack sequence of type* 1 *is* $(1, 2)$ *of attacking target* 1 *at step* 1 *and attacking target* 2 *at step* 2*. The attack sequence of type* 2 *is* $(2, 2)$*.*

Deceptive attacker. *Let's consider what happens if the attacker plays deceptively. First, Type* 1 *has no incentive to play deceptively. In fact, if Type 1 pretends to be type* 2 *(which means type 1 will play the attack sequence* $(2, 2)$*, Type* 1 *would receive a total expected utility of* $(8\times(1-0.7647)+(-8)\times 0.7647)+(8\times(1-0.7647)+(-8)\times 0.7647) = -8.4704 < 1.0588$ *where* 1.0588 *is the total expected utility of Type* 1 *if he plays non-deceptively.*

On the other hand, Type 2 *would like to pretend to be Type* 1*. Indeed, if Type* 2 *mimics Type* 1 *(which means Type* 2 *would play the attack sequence* $(1, 2)$*), the total expected utility which Type* 2 *would receive is:* $(3\times(1-0.2353)+(-4)\times 0.2353)+(9\times(1-0.5)+(-1)\times 0.5) = 5.3529 > 2.706$ *where* 2.706 *is the total expected utility that Type* 2 *would receive if playing non-deceptively.*

As a result, since both attacker types play the attack sequence of $(1, 2)$*, the defender would receive a total utility of* $(2\times 0.2353 + (-8)\times(1-0.2353)) + 6\times 0.5 + (-10)\times(1-0.5) = -7.647$ *which is much less than* -1.5883*.*

5 Defense against Imitative Attacks

To address the challenge of the attacker's imitative deception, we design a carefully-tuned defender strategy plan that takes into account the attacker's deception. The goal of the defender is to find an optimal strategy plan that maximizes the defender's total expected utility. The problem can be formulated as the following Mixed Integer Quadratic Program (MIQP) which is an extension of the MILP (2-9):

$$\max \sum\nolimits_{\lambda'} \Big(\sum\nolimits_{\lambda} p_\lambda q(\lambda,\lambda')\Big) \sum\nolimits_{t=1}^{T} EU^{d,t}_{\lambda'} \tag{10}$$

$$\text{s.t. } U^\lambda \leq \sum\nolimits_{t=1}^{T} EU^{\lambda,t}_{\lambda'} + (1-q(\lambda,\lambda'))M, \quad \forall\lambda,\lambda' \tag{11}$$

$$U^\lambda \geq \sum\nolimits_{t=1}^{T} EU^{\lambda,t}_{\lambda'}, \quad \forall\lambda,\lambda' \tag{12}$$

$$EU^{\lambda,t}_{\lambda'} \leq EU^\lambda(\mathbf{x}^{\lambda'}_t, i) + (1 - z^{\lambda'}_t(i))M, \quad \forall\lambda,\lambda',t,i \tag{13}$$

$$EU^{\lambda,t}_{\lambda'} \geq EU^\lambda(\mathbf{x}^{\lambda'}_t, i) - (1 - z^{\lambda'}_t(i))M, \quad \forall\lambda,\lambda',t,i \tag{14}$$

$$\sum\nolimits_{\lambda'} q(\lambda,\lambda') = 1, q(\lambda,\lambda') \in \{0,1\}, \quad \forall(\lambda,\lambda') \tag{15}$$

Constraints (3-9) are satisfied. (16)

where the objective is non-linear. In addition to the variables from (2-9), this MIQP has the following additional variables:

- $q(\lambda,\lambda')$: represents the deception choice of type λ; $q(\lambda,\lambda') = 1$ if type λ pretends to be type λ' and $q(\lambda,\lambda') = 0$, otherwise.
- U^λ: is the optimal total expected utility of type λ.
- $EU^{\lambda,t}_{\lambda'}$: is the immediate expected utility of type λ at step t if this type pretends to be type λ'.

Constraints (11–12) ensure each type λ chooses the best deceptive type to mimic. Constraints (13–14) guarantee that if type λ pretends to be type λ', the type λ will receive an immediate expected utility at each t which depends on (i) the defense strategy that type λ' faces at t; and (ii) the corresponding myopic best response of type λ'.

In the objective (10), the factor $\big(\sum_\lambda p_\lambda q(\lambda,\lambda')\big)$ is essentially the probability the defender faces the myopic-best responses of type λ'. Therefore, the objective (10) is the defender's total expected utility and the MIQP (10–16) returns an optimal defender strategy plan.

Mixed Integer Linear Program Conversion. To convert this MIQP into a MILP, we introduce a new variable, $v(\lambda,\lambda') = q(\lambda,\lambda')\sum_{t=1}^{T} EU^{d,t}_{\lambda'}$, with the following additional constraints:

$$v(\lambda,\lambda') \leq \sum_{t=1}^{T} EU^{d,t}_{\lambda'} + (1-q(\lambda,\lambda'))M, \quad \forall\lambda,\lambda' \tag{17}$$

$$v(\lambda,\lambda') \geq \sum_{t=1}^{T} EU^{d,t}_{\lambda'} - (1-q(\lambda,\lambda'))M, \quad \forall\lambda,\lambda' \tag{18}$$

$$v(\lambda,\lambda') \geq -q(\lambda,\lambda')M, \quad \forall\lambda,\lambda' \tag{19}$$

$$v(\lambda,\lambda') \leq q(\lambda,\lambda')M, \quad \forall\lambda,\lambda' \tag{20}$$

where constraints (17–18) guarantee that $v(\lambda,\lambda') = \sum_{t=1}^{T} EU^{d,t}_{\lambda'}$ if type λ pretends to be type λ' ($q(\lambda,\lambda') = 1$). On the other hand, constraints (19–20) ensure that $v(\lambda,\lambda') = 0$ if $q(\lambda,\lambda') = 0$.

Given the new variables $\{v(\lambda,\lambda')\}$, we can convert the non-linear objective in (10) to the following linear objective:

$$\sum_{\lambda'}\sum_{\lambda} p_\lambda v(\lambda,\lambda')$$

Overall, solving (10–16) is computationally expensive since the MILP involves an exponential number of attack histories and future possibilities. Therefore, we propose the following two accelerated heuristics: (i) limited-history-length heuristic — we limit the length of historical attack sequences on which the defender's strategies depend on. That is, the defender's strategy at each step t only depends on a truncated attack history $\mathbf{h}_{(t-\delta T+1):t} = \{i_{t-\delta T}, \ldots, i_{t-1}\}$ where δT is the length of historical attacks in consideration; and (ii) limited-look-ahead heuristic — we limit the number of future time steps to consider. The two heuristics are elaborated in next sections.

5.1 Limited-History-Length Heuristic

Given a limited history length δT, the problem of finding an optimal strategy plan for the defender can be represented as a MILP similar to (10–16) excepts the constraints (6 – 7). In particular, we replace these two constraints by the following two constraints: $\forall \mathbf{h}_{(t-\delta T+1):t}$,

$$x^{\lambda'}_t(i) \leq x_i(t, \mathbf{h}_{(t-\delta T+1):t}) + (\delta T - \sum_{t-\delta T \leq t' \leq t-1} z^{\lambda'}_{t'}(i_{t'}))M$$

$$x^{\lambda'}_t(i) \geq x_i(t, \mathbf{h}_{(t-\delta T+1):t}) - (\delta T - \sum_{t-\delta T \leq t' \leq t-1} z^{\lambda'}_{t'}(i_{t'}))M$$

Here, $\{x_i(t, \mathbf{h}_{(t-\delta T+1):t})\}$ is the defender strategy at step t with respect to the truncated history $\mathbf{h}_{(t-\delta T+1):t}$. The above two constraints guarantee that the strategy $\{x^{\lambda'}_t(i)\}$ which the attacker type λ' faces at step t is the defender strategy $\{x_i(t, \mathbf{h}_{(t-\delta T+1):t})\}$ if the sequence of best responses of type λ' in the previous δT steps matches the truncated attack history $\mathbf{h}_{(t-\delta T+1):t} = \{i_{t-\delta T}, \ldots, i_{t-1}\}$ (i.e., $z^{\lambda'}_{t'}(i_{t'}) = 1$ for all $t' \in \{t-\delta T, \ldots, t-1\}$).

5.2 Limited-Look-Ahead Heuristic

The overview of the limited-look-ahead heuristic is illustrated in Algorithm 1. Essentially, Algorithm 1 divides the time horizon $\{1, 2, \dots, T\}$ into multiple time blocks; each block consists of $\delta T_0 + 1$ time steps: $\{1, \dots, \delta T_0 + 1\}, \{\delta T_0 + 2, \dots, 2 * \delta T_0 + 2\} \dots$. The algorithm then iteratively computes the defender's strategies within each time block based on a modification of the `MILP` (10 – 16). In this modified `MILP`, the total number of time steps is set to that time block only while the defender strategies in previous time blocks are pre-computed in the previous iteration of the algorithm.

Algorithm 1: Limited-Look-Ahead Heuristic

```
1 $T$: total number of steps, $\delta T_0$: limited number of future steps;
2 Initialize $t = 1$ and partial strategy plan $\pi^*_{1\dots t-1} = \emptyset$;
3 while $t \leq T$ do
4     Set $\delta T = \delta T_0$;
5     if $t + \delta T_0 > T$ then $\delta T = T - t$;
6     Solve MILP (10–16) with a fixed partial plan $\pi^*_{1\dots t-1}$ and
        total $\delta T + 1$ steps $(t, \dots, t + \delta T)$ to consider;
7     Obtain $\pi^*_{1\dots t+\delta T} = \pi^*_{1\dots t-1} \cup \{x^*_i(t', \mathbf{h}_{t'})\}_{t'=t}^{t+\delta T}$;
8     Update $t = t + \delta T + 1$;
```

Since the strategies in previous time blocks are precomputed and fixed, it results in fixed sequences of best responses of attacker types prior to the current time blocks. As a result, the total number of attack histories considered in the modified `MILP` is $O(\Lambda \times N^{\delta T_0})$.

6 Experiments

In our experiments, we evaluate both the solution quality and runtime performance of our proposed algorithms. We focus on analyzing the loss of the defender and the benefit of the attacker in terms of expected utility as the result of the attacker deception. The rewards and penalties of the players are generated uniformly at random within the range of $[1, 10]$ and $[-10, -1]$, respectively. The prior distribution over the attacker types are generated uniformly at random. We examine games with (i) three attacker types (2, 3, and 4 time steps); and four attacker types (2 and 3 time steps). We compare the players' utilities for playing strategies computed in three scenarios:

- The attacker is deceptive and the defender takes into account the attacker's deception. Five algorithms are evaluated: `Optimal` — the players' strategies are computed by the exact algorithm (10–16); `LA 1-Step` and `LA 2-Step` — the players' strategies are generated based on the limited-look-ahead heuristic with 1 step and 2 steps to look ahead, respectively; and `LM 1-Step` and `LM 2-Step` — the limited-memory-length heuristic is used with 1 step and 2 steps of observation history to consider.
- The attacker is non-deceptive and the defender also assumes so (`Non-Dec`). The defender's strategies are generated based on the `MILP` introduced in Section 3.2.
- The attacker is deceptive while the defender assumes the attacker is non-deceptive (`Dec-Unaddressed`).

6.1 Solution quality

In our first set of experiments, we examine various security game settings with 3 attacker types. For each game setting (i.e., a tuple (N, K, T) of the number of targets, the number of security resources, the number of time steps), we generates 100 game instances.

We first examine which game instances in which the attacker has incentive to play deceptively. Note that, the attacker's incentive comes from obtaining a higher expected utility for playing deceptively. Otherwise, the attacker would play truthfully. We summarize our result in Table 3 which shows the percentage of the game instances in which the attacker plays deceptively for its benefit. In Table 3, the percentage of games in which the attacker is deceptive increases when the number of steps increases. This result reflects that the attacker is more likely to be deceptive for a longer-term benefit.

2 steps	3 steps	4 steps
34%	57%	61%

Table 3: Percentage of games in which the attacker is deceptive.

We next examine the loss of the defender and benefit of the attacker when the attacker plays deceptively. The results are shown in Figures 1 and 2. The x-axis is the number of targets in the games and the y-axis is the defender's expected utility (Figure 1) or the attacker's expected utility (Figure 2) on average. Note that in 2-step and 3-step games, `LA 2-step` and `Optimal` are exactly the same since `LA 2-step` considers all three steps of the games. Therefore, in 3-step games, we do not include the results of `Optimal` for the sake of representation. Finally, in 4-step games, `LM 1-step`, `LM 2-step`, and `Optimal` are too computationally expensive. Therefore, we only compare between `LA 1-step`, `LA 2-step`, `Non-Dec`, and `Dec-Unaddressed`.

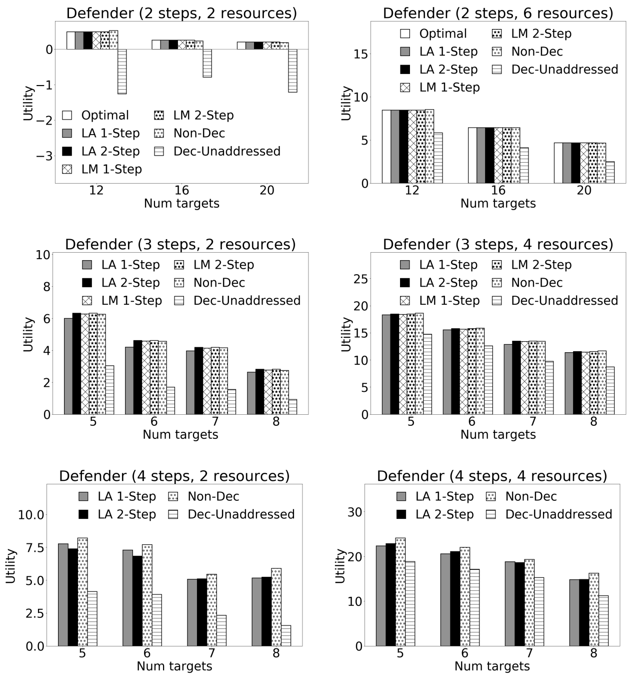


Figure 1: Analysis on defender's averaged utility in 3-attacker games

Figure 1 shows that the defender suffers a significant loss when he does not address the attacker's deception (`Non-Dec` versus `Dec-Unaddressed`). Conversely, the defender strategies generated by our counter-deception algorithms lead to a significant increase in the defender's utility (`Optimal`, `LM 1-Step`, `LM 2-Step`, `LA 1-Step`, and `LA 2-Step`). In fact, the defender's utility for addressing the attacker deception is approximately the same as the non-deceptive case in most game settings. This result shows that our counter-deception

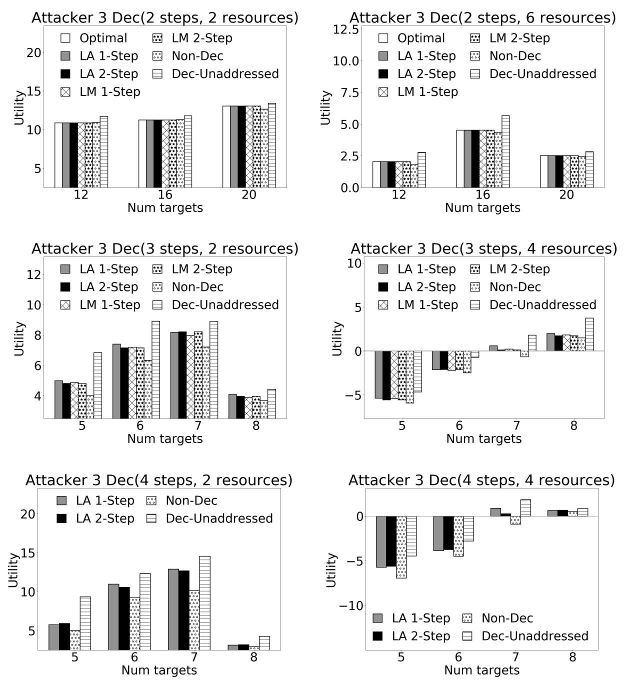


Figure 2: Analysis on attacker's average utility in 3-attacker games

algorithms helps in diminishing drastically the impact of the attacker's deception. In addition, our approximate heuristics provides the defender's utility close to the exact algorithm.

Figure 2 shows the impact of the attacker's deception on the attacker's utility. The attacker receives a substantially higher expected utility for playing deceptively when the defender does not handle the attacker's deceit (`Dec-Unaddressed` versus `Non-Dec`). However, when the defender takes the attacker's deception into account (`Optimal`, `LM 1-Step`, `LM 2-Step`, `LA 1-Step`, and `LA 2-Step`), the attacker benefit for being deceptive is reduced significantly.

Finally, Figure 3 shows the defender and attacker's utility in games with 4 attacker types. We compare between `LA 1-Step`, `LA 2-Step`, `Non-Dec`, and `Dec-Unaddressed`. Note that in 2-step and 3-step games, `LA 2-Step` and `Optimal` are equivalent. In Figure 3, we also observe a similar trend in the loss of the defender and the benefit of the attacker compared with the 3-attacker case.

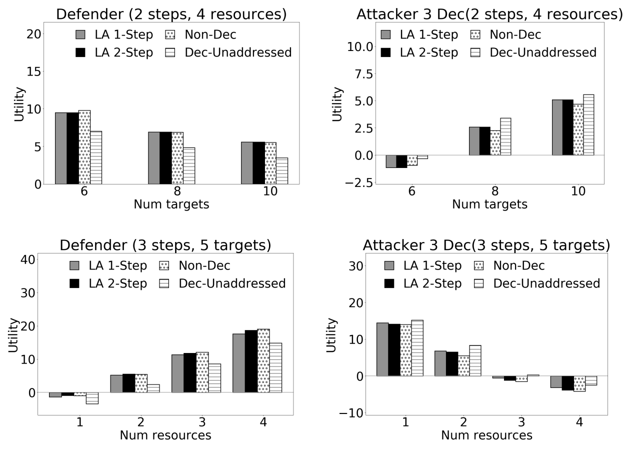
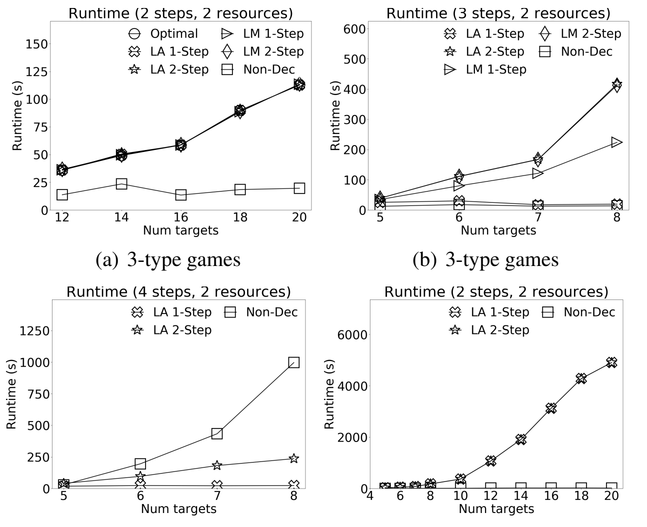


Figure 3: Analysis on players' average utility in 4-attacker games



(a) 3-type games (b) 3-type games

(c) 3-type games (d) 4-type games

Figure 4: Runtime performance

6.2 Runtime performance

Our evaluation on runtime performance is shown in Figure 4. The x-axis is the number of targets and the y-axis is the averaged runtime in seconds. Overall, the runtime of our counter-deception algorithms grows gradually when the number of targets increases. The `Non-Dec` algorithm (to deal with a non-deceptive attacker) has the best runtime performance compared to all other algorithms (for addressing the attacker's deception). In 2-step games, `Optimal`, `LM 1-Step`, `LM 2-Step`, `LA 1-Step`, and `LA 2-Step` are identical. Therefore, the runtime performance of these algorithms is the same in Figure 4(a). In 3-step games (Figure 4(b)) and 4-step games (Figure 4)(c), among the evaluated counter-deception algorithms, `LA 1-Step` runs significantly faster compared to the other algorithms. In consideration of the trade-off between solution quality and runtime performance, `LA 1-Step` is shown to be the best to solve games with a large T.

7 Summary

In this work, we focus on addressing the challenge of attacker deception in `RBSSGs`. We study the imitative attacker deception — a basic deception strategy of the attacker in which the attacker mimics a type (which is different from its true type) and consistently plays the game according to that deceptive type. We have four main contributions. First, we present a new exact algorithm to compute an equilibrium of standard `RBSSGs`, when the attacker is not manipulative. Second, built on this equilibrium computation, we introduce a new exact counter-deception algorithm to address the attacker's imitative deception. Third, we present two new heuristics: limited-look-ahead and limited-memory-length, to deal with the computation challenge of exponentially many observation histories and future possibilities. Finally, our extensive experiments show a significant benefit of the attacker and significant loss of the defender when the attacker deception is not addressed. Our counter-deception algorithm helps in drastically reducing the impact of the attacker deception.

REFERENCES

[1] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia, 'Commitment without regrets: Online learning in Stackelberg security games', in *16th ACM Conference on Economics and Computation*, pp. 61–78, (2015).

[2] Marco Barreno, Blaine Nelson, Anthony D. Joseph, and J. D. Tygar, 'The security of machine learning', *Machine Learning*, **81**(2), 121–148, (2010).

[3] Avrim Blum, Nika Haghtalab, and Ariel D. Procaccia, 'Learning optimal commitment to overcome insecurity', in *Advances in Neural Information Processing Systems*, pp. 1826–1834, (2014).

[4] Fei Fang, Thanh H Nguyen, Rob Pickles, Wai Y Lam, Gopalasamy R Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux, 'Deploying paws: Field optimization of the protection assistant for wildlife security', in *Twenty-Eighth IAAI Conference*, (2016).

[5] D. Fraunholz, S. D. Anton, C. Lipps, D. Reti, D. Krohmer, F. Pohl, M. Tammen, and H. D. Schotten, 'Demystifying deception technology: A survey', *arXiv preprint arXiv:1804.06196*, (2018).

[6] Jiarui Gan, Qingyu Guo, Long Tran-Thanh, Bo An, and Michael Wooldridge, 'Manipulating a learning defender and ways to counteract', in *Advances in Neural Information Processing Systems*, (2019).

[7] Jiarui Gan, Haifeng Xu, Qingyu Guo, Long Tran-Thanh, Zinovi Rabinovich, and Michael Wooldridge, 'Imitative follower deception in stackelberg games', in *Proceedings of the 2019 ACM Conference on Economics and Computation*, EC '19, pp. 639–657, New York, NY, USA, (2019). ACM.

[8] S. Gholami, B. Ford, F. Fang, A. Plumptre, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, M. Nsubaga, and J. Mabonga, 'Taking it for a test drive: a hybrid spatio-temporal model for wildlife poaching prediction evaluated through a controlled field test', in *European Conference on Machine Learning*, (2017).

[9] Qingyu Guo, Bo An, Branislav Bosansky, and C. Kiekintveld, 'Comparing strategic secrecy and Stackelberg commitment in security games', in *26th International Joint Conference on Artificial Intelligence*, (2017).

[10] Nika Haghtalab, Fei Fang, Thanh Hong Nguyen, Arunesh Sinha, Ariel D. Procaccia, and Milind Tambe, 'Three strategies to success: Learning adversary models in security games.', in *25th International Joint Conference on Artificial Intelligence*, pp. 308–314, (2016).

[11] D. Kar, B. Ford, S. Gholami, F. Fang, A. Plumptre, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, and M. Nsubaga, 'Cloudy with a chance of poaching: Adversary behavior modeling and forecasting with real-world poaching data', in *16th International Conference on Autonomous Agents and Multi-Agent Systems*, (2017).

[12] Christian Kroer and Tuomas Sandholm, 'Limited lookahead in imperfect-information games', in *Twenty-Fourth International Joint Conference on Artificial Intelligence*, (2015).

[13] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala, 'Learning and approximating the optimal strategy to commit to', in *International Symposium on Algorithmic Game Theory*, pp. 250–262, (2009).

[14] Daniel Lowd and Christopher Meek, 'Adversarial learning', in *ACM SIGKDD International Conference on Knowledge Discovery in Data Mining*, pp. 641–647, (2005).

[15] Janusz Marecki, Gerry Tesauro, and Richard Segal, 'Playing repeated Stackelberg games with unknown opponents', in *11th International Conference on Autonomous Agents and Multiagent Systems*, (2012).

[16] T. H. Nguyen, A. Sinha, S. Gholami, A. Plumptre, L. Joppa, M. Tambe, M. Driciru, F. Wanyama, A. Rwetsiba, R. Critchlow, et al., 'Capture: A new predictive anti-poaching tool for wildlife protection', in *15th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 767–775, (2016).

[17] Thanh Nguyen and Haifeng Xu, 'Imitative attacker deception in stackelberg security games', in *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pp. 528–534. International Joint Conferences on Artificial Intelligence Organization, (7 2019).

[18] Thanh H Nguyen, Yongzhao Wang, Arunesh Sinha, and Michael P Wellman, 'Deception in finitely repeated security games', in *33th AAAI Conference on Artificial Intelligence*, (2019).

[19] B. Peng, Weiran Shen, Pingzhong Tang, and Song Zuo, 'Learning optimal strategies to commit to', in *33th AAAI Conference on Artificial Intelligence*, (2019).

[20] Zinovi Rabinovich, Albert Xin Jiang, Manish Jain, and Haifeng Xu, 'Information disclosure as a means to security', in *14th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 645–653, (2015).

[21] Arunesh Sinha, Fei Fang, Bo An, Christopher Kiekintveld, and Milind Tambe, 'Stackelberg security games: Looking beyond a decade of success.', in *IJCAI*, pp. 5494–5501, (2018).

[22] Milind Tambe, *Security and game theory: algorithms, deployed systems, lessons learned*, Cambridge University Press, 2011.

[23] Haifeng Xu, Zinovi Rabinovich, Shaddin Dughmi, and Milind Tambe, 'Exploring information asymmetry in two-stage security games.', in *29th AAAI Conference on Artificial Intelligence*, pp. 1057–1063, (2015).

[24] Haifeng Xu, Long Tran-Thanh, and Nicholas R. Jennings, 'Playing repeated security games with no prior knowledge', in *15th International Conference on Autonomous Agents and Multi-Agent Systems*, pp. 104–112, (2016).

[25] Jun Zhuang, Vicki M. Bier, and Oguzhan Alagoz, 'Modeling secrecy and deception in a multi-period attacker-defender signaling game', *European Journal of Operational Research*, **203**, 409–418, (2010).