2-2015

Commitment Without Regrets: Online Learning in Stackelberg Security Games

Maria-Florina Balcan
Carnegie Mellon University, ninamf@cs.cmu.edu

Avrim Blum
Carnegie Mellon University, avrim@cs.cmu.edu

Nika Haghtalab
Carnegie Mellon University

Ariel D. Procaccia
Carnegie Mellon University, arielpro@cs.cmu.edu

A

Commitment Without Regrets: Online Learning in Stackelberg Security Games

Maria-Florina Balcan, Carnegie Mellon University
Avrim Blum, Carnegie Mellon University
Nika Haghtalab, Carnegie Mellon University
Ariel D. Procaccia, Carnegie Mellon University

In a *Stackelberg Security Game*, a *defender* commits to a randomized deployment of security resources, and an *attacker* best-responds by attacking a target that maximizes his utility. While algorithms for computing an optimal strategy for the defender to commit to have had a striking real-world impact, deployed applications require significant information about potential attackers, leading to inefficiencies. We address this problem via an online learning approach. We are interested in algorithms that prescribe a randomized strategy for the defender at each step against an adversarially chosen sequence of attackers, and obtain feedback on their choices (observing either the current attacker type or merely which target was attacked). We design *no-regret algorithms* whose regret (when compared to the best fixed strategy in hindsight) is polynomial in the parameters of the game, and sublinear in the number of times steps.

Categories and Subject Descriptors: I.2.11 [**Distributed Artificial Intelligence**]: Multiagent Systems; J.4 [**Computer Applications**]: Social and Behavioral Sciences—*Economics*

Additional Key Words and Phrases: No-regret learning, Stackelberg security games

1. INTRODUCTION

A *Stackelberg game* includes two players — the *leader* and the follower. The leader plays first by *committing* to a mixed strategy. The leader's commitment is observed by the follower, who then plays a *best response* to the leader's strategy.

When Heinrich Freiherr von Stackelberg published his model of competing firms in 1934, he surely did not imagine that he is laying the foundations for an exciting real-world application of *computational* game theory: *Stackelberg Security Games (SSGs)*. On a conceptual level, the role of the leader in an SSG is typically played by a security agency tasked with the protection of critical infrastructure sites; it is referred to as the *defender*. The defender's strategy space consists of assignments of resources to potential targets. The follower — referred to as the *attacker* in this context — observes the leader's (possibly randomized) deployment, and chooses a target to attack in a way that maximizes his own expected utility. Algorithms that help the defender compute an optimal strategy to commit to are currently in use by major security agencies such as the US Coast Guard, the Federal Air Marshals Service, and the Los Angeles Airport Police; see the book by Tambe [2012] for an overview. Strikingly, similar algorithms are currently being prepared for deployment in a completely different domain: wildlife protection in Uganda and Malaysia [Yang et al. 2014].

These impressive applications have motivated a rapidly growing body of work devoted to understanding and overcoming the limitations of the basic SSG model. Here we focus on the model's arguably most severe limitation: in order to compute the optimal strategy to commit to, the leader must know the attacker's utility function. In practice, estimates are obtained from risk analysis experts and historical data, but the degree of uncertainty (and, therefore, inefficiency) is typically high.

Several papers [Letchford et al. 2009; Marecki et al. 2012; Blum et al. 2014b] alleviate uncertainty by *learning* an optimal strategy for the defender. Learning takes place through repeated interaction with the attacker: at each round, the defender commits to a mixed strategy, and observes the attacker's best response; one can think of this process as learning with *best-response queries*. But (as discussed in more detail in Section 1.3) these papers are restricted to a repeated interaction with a single attacker type [Marecki et al. 2012; Blum et al. 2014b], or simple variations thereof [Letchford

Authors' addresses: School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, 15217, email: {ninamf,avrim,nhaghtal,arielpro}@cs.cmu.edu. This work was partially supported by the NSF under grants CCF-1215883 and IIS-1350598.

et al. 2009]. Either way, the defender faces the exact same situation in each round — a convenient fact that allows the learning process to ultimately converge to an (almost) optimal strategy, which is then played until the end of time.

1.1. Model and Conceptual Contributions

In this paper we deal with uncertainty about attackers by adopting a fundamentally different approach, which makes a novel connection to the extensive literature on *online learning*. Similarly to previous work [Letchford et al. 2009; Marecki et al. 2012; Blum et al. 2014b], we study a repeated Stackelberg game; but in our setting the attackers are not all the same — in fact, they are chosen *adversarially* from some known set K. That is, at each round the defender commits to a mixed strategy based on the history of play so far, and an adversarially chosen attacker from K best-responds to that strategy.

Even in the face of this type of uncertainty, we would like to compete with the best fixed mixed strategy *in hindsight*, that is, the mixed strategy that yields the highest total payoff when played against each attacker in the sequence generated by the adversary. The *regret* associated with an online learning algorithm (which recommends to the defender a mixed strategy at each step) is simply the difference between the utility of the best-in-hindsight fixed strategy and the expected utility of the algorithm in the online setting. Our goal is to

> *... design online learning algorithms whose regret is sublinear in the number of time steps, and polynomial in the parameters of the game.*

Such an algorithm — whose average regret goes to zero as the number of time steps goes to infinity — is known as a *no-regret algorithm*. While there has been substantial work on no-regret learning, what makes our situation different is that our goal is to compete with the best mixed strategy in hindsight against the sequence of attackers that arrived, *not* the sequence of targets they attacked.

1.2. Overview of Our Results

We provide two algorithmic results that apply to two different models of feedback. In the *full information* model (Section 5), the defender plays a mixed strategy, and observes the type of attacker that responds. This means that the algorithm can infer the attacker's best response to *any* mixed strategy, not just the one that was played. We design an algorithm whose regret is $O(\sqrt{Tn^2k\log nk})$, where T is the number of time steps, n is the number of targets that can be attacked, and k is the number of attacker types.

In the second model — the *partial information* model (Section 6) — the defender only observes which target was attacked at each round. Our main technical result is the design and analysis of a no-regret algorithm in the partial information model whose regret is bounded by $O(T^{2/3}nk\log^{1/3} nk)$.

For both results we assume that the attackers are selected (adversarially) from a set of k *known* types. It is natural to ask whether no-regret algorithms exist when there are no restrictions on the types of attackers. In Section 7, we answer this question in the negative, thereby justifying the dependence of our bounds on k.

Let us make two brief remarks regarding central issues that are discussed at length in Section 8. First, throughout the paper we view *information*, rather than *computation*, as the main bottleneck, and therefore aim to minimize regret without worrying (for now) about computational complexity. Second, our exposition focuses on Stackelberg *Security* Games, but our framework and results apply to Stackelberg games more generally.

1.3. Related work

To the best of our knowledge, only three previous papers take a learning-theoretic approach to the problem of unknown attackers in SSGs (and none study an online setting):

(1) Blum et al. [2014b] study the case of a repeated interaction with a *single unknown attacker*; they design an algorithm that learns an almost optimal strategy for the defender using a polynomial number of best-response queries.
(2) Letchford et al. [2009] also focus on interaction with a single unknown attacker. Their algorithm learns the attacker's actual utility function on the way to computing an optimal strategy for the defender; it requires a polynomial number of queries in the explicit representation of the given *Stackelberg game*, which could be exponential in the size of the Stackelberg *security* game. They note that their results can be extended to *Bayesian* Stackelberg games, where there is a known distribution over attacker types: to learn the utility function of a each attacker type, they simply run the single-attacker learning algorithm but repeat each best response query until the response of the desired attacker type is observed. This simple reduction is incompatible with the techniques of Blum et al. [2014b].
(3) Marecki et al. [2012] discuss Bayesian Stackelberg games, but, again, concentrate on the case of a single attacker type (which is initially drawn from a distribution but then played repeatedly). Similarly in spirit to our work, they are interested in the *exploration-exploitation tradeoff*; their algorithm is shown empirically to provide good guarantees in the short term, and provably converges to the optimal strategy in the long term. No other theoretical guarantees are given; in particular the algorithm's convergence time is unknown.

It is also worth mentioning that other papers have taken different approaches to dealing with attacker uncertainty in the context of SSGs, e.g., robust optimization [Pita et al. 2010] and representing utilities via continuous distributions [Kiekintveld et al. 2011].

Our paper is also closely related to work on online learning. For the full information feedback case, the seminal work of Littlestone and Warmuth [1994] achieves a regret bound of $O(\sqrt{T \log N})$ for N strategies. Kalai and Vempala [2005] show that when the set of strategies is a subset of $\mathbb{R}^d$ and the loss function is linear, this bound can be improved to $O(\sqrt{T \log d})$. This bound is applicable when there are infinitely many strategies. In our work, the space of mixed strategies can be translated to a n-dimensional space, where n is the number of targets. However, our loss function depends on the best response of the attackers, hence is not linear.

For the partial feedback case (also known as the *bandit* setting), Auer et al. [1995] introduce an algorithm that achieves regret $O(\sqrt{TN \log N})$ for N strategies. Awerbuch and Kleinberg [2008] extend the work of Kalai and Vempala [2005] to the partial information feedback setting for online routing problems, where the feedback is in the form of the end-to-end delay of a chosen source-to-sink path. Their approach can accommodate exponentially or infinitely large strategy spaces, but again requires a linear loss function.

2. PRELIMINARIES

We consider a repeated Stackelberg security game between a defender (the leader) and a sequence of attackers (the followers). At each step of this repeated game, the interactions between the defender and the attacker induce a Stackelberg security game, where the defender commits to a randomized allocation of his security resources to defend potential targets, and the attacker, in turn, observes this randomized allocation and attacks the target with the best expected payoff. The defender and the attacker then receive payoffs. The defender's goal is to maximize his payoff over a period of time, even when the sequence of attackers is unknown.

More precisely, a *repeated stackelberg security game* includes the following components:

- *Time horizon* T: the number of rounds.
- Set of *targets* $N = \{1, \ldots, n\}$.
- A defender with:
  - *Resources*: A set of *resources* R.
  - *Schedules:* A collection $\mathcal{D} \subseteq 2^N$ of schedules. Each schedule $D \in \mathcal{D}$ represents a set of targets that can be simultaneously defended by one resource.

— *Assignment function*: Function $A : R \to 2^{\mathcal{D}}$ indicates the set of all schedules that can be defended by a given resource. An assignment of resources to schedules is *valid* if every resource r is allocated to a schedule in $A(r)$.
— *Strategy*: A *pure strategy* is a valid assignment of resources to schedules. The set of all pure strategies is determined by $N, \mathcal{D}, R$, and A and can be represented as follow. Let there be m pure strategies, and let M be a zero-one $n \times m$ matrix, such that the rows represent targets and columns represent pure strategies, with $M_{i,j} = 1$ if and only if target i is covered by some resource in the j^{th} pure strategy.
A *mixed strategy* is a distribution over the set of pure strategies and is represented by an $m \times 1$ probability vector $\mathbf{s}$, such that for all j, s_j is the probability with which pure strategy j is played. Every mixed strategy induces a *coverage probability* vector $\mathbf{p} \in [0, 1]^n$, where p_i is the probability with which target i is defended under that mixed strategy. The mapping between a mixed strategy, $\mathbf{s}$, and its coverage probability vector, $\mathbf{p}$, is given by $\mathbf{p} = M\mathbf{s}$.

◦ A set of all attacker types $K = \{\alpha_1, \ldots, \alpha_k\}$.
— An adversary selects a *sequence of attackers* $\mathbf{a} = a_1, \ldots, a_T$, such that for all t, $a_t \in K$.
— Throughout this work we assume that K is known to the defender, while $\mathbf{a}$ remains unknown.

◦ *Utilities*: The defender and attacker both receive payoffs when a target is attacked.
— For the defender, let $u_d^c(i)$ and $u_d^u(i)$ be the defender's payoffs when target i is attacked and is, respectively, covered or not covered by the defender. Then, under coverage probability $\mathbf{p}$, the expected utility of the defender when target i is attacked is given by $U_d(i, \mathbf{p}) = u_d^c(i)p_i + u_d^u(i)(1 - p_i)$. Note that $U_d(i, \mathbf{p})$ is linear in $\mathbf{p}$. All utilities are normalized such that $u_d^c(i), u_d^u(i) \in [-1, 1]$ for all $i \in N$.
— For an attacker of type α_j, let $u_{\alpha_j}^c(i)$ and $u_{\alpha_j}^u(i)$ be the attacker's payoffs from attacking target i when the target is, respectively, covered or not covered by the defender. Then under coverage probability $\mathbf{p}$, the expected utility of the attacker from attacking target i is given by $U_{\alpha_j}(i, \mathbf{p}) = u_{\alpha_j}^c(i)p_i + u_{\alpha_j}^u(i)(1 - p_i)$. Note that $U_{\alpha_j}(i, \mathbf{p})$ is linear in $\mathbf{p}$.

At step t of the game, the defender chooses a mixed strategy to deploy over his resources. Let $\mathbf{p}_t$ be the coverage probability vector corresponding to the mixed strategy deployed at step t. Attacker a_t observes $\mathbf{p}_t$ and *best-responds* to it by attacking target $b_{a_t}(\mathbf{p}) = \arg\max_{i \in N} U_{a_t}(i, \mathbf{p})$. When multiple targets have the same payoff to the attacker, each attacker breaks ties in some arbitrary but consistent order.

Note that given a coverage probability vector, the utilities of all players and the attacker's best-response is invariant to the mixed strategy that is used to implement that coverage probability. Therefore, we work directly in the space of valid coverage probability vectors, denoted by $\mathcal{P}$. For ease of exposition, in the remainder of this work we do not distinguished between a mixed strategy and its coverage probability vector.

3. PROBLEM FORMULATION

In this section, we first formulate the question of *how a defender can effectively protect targets in a repeated Stackelberg security game when the sequence of attackers is not known to him.* We then give an overview of our approach.

Consider a repeated Stackelberg security setting with one defender and a sequence of attackers, $\mathbf{a}$, that is selected beforehand by an adversary. When $\mathbf{a}$ is not known to the defender a priori, the defender has to adopt an online approach to maximize his overall payoff during the game. That is, at every time step t the defender plays, possibly at random, a mixed strategy $\mathbf{p}_t$ and receives some "feedback" regarding the attacker at that step. The defender subsequently adjusts his mixed strategy $\mathbf{p}_{t+1}$ for the next time step. The expected payoff of the defender is given by

$$\mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right],$$

where, importantly, the expectation is taken only over internal randomness of the defender's online algorithm.

The feedback the defender receives at each time step plays a major role in the design of the algorithm. In this work, we consider two types of feedback, *full information* and *partial information*. In the full information case, after attacker a_t best-responds to $\mathbf{p}_t$ by attacking a target, the defender observes a_t, that is, the defender know which attacker type just attacked. In the partial information case, even after the attack occurs the defender only observes the target that was attacked, i.e. $b_{a_t}(\mathbf{p}_t)$. More formally, in the full-information and partial-information settings, the choice of $\mathbf{p}_t$ depends on a_i for all $i \in [t-1]$ or $b_{a_i}(\mathbf{p}_i)$ for all $i \in [t-1]$, respectively. As a sanity check, note that knowing a_i is sufficient to compute $b_{a_i}(\mathbf{p}_i)$, but multiple attacker types may respond by attacking the same target; so feedback in the full information case is indeed strictly more informative.

To examine the performance of our online approach, we compare its payoff to the defender's payoff in a setting where $\mathbf{a}$ is known to the defender. In the event that the sequence of attackers, or merely the frequency of each attacker type in the sequence, is known, the defender can pre-compute a fixed mixed strategy with the best payoff against that sequence of attackers and play it at every step of the game. We refer to this mixed strategy as the *best mixed strategy in hindsight* and denote it by

$$\mathbf{p}^* = \arg\max_{\mathbf{p}} \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}), \mathbf{p}).$$

Our goal is to design online algorithms for the defender with payoff that is almost as good as the payoff of the best mixed strategy in hindsight. We refer to the difference between the utility of the online algorithm and the utility of the best-in-hindsight mixed strategy as *regret*, i.e.,

$$\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - \mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right].$$

Our results are stated as upper and lower bounds on regret. For upper bounds, we show both in the case of full information feedback (Theorem 5.1) and partial information feedback (Theorem 6.1) that

$$\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - \mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right] \leq o(T) \cdot \mathrm{poly}(n, k).$$

In particular, the average regret goes to zero as T goes to infinity, that is, our algorithms are *no-regret algorithms*. In contrast, we show that even in the full-information case, when K is large compared to T, our regret must be linear in T (Theorem 7.1).

3.1. Methodology

This formulation of our problem, which involves comparison between online and offline decision making, closely matches (by design) the classic online learning framework. To formally introduce this setup, consider a set of actions $\mathcal{M}$, time horizon T, an online learner, and an *adaptive* adversary. For every $t \in [T]$, the learner chooses a distribution $\mathbf{q}_t$ over the actions in $\mathcal{M}$, and then picks a random action based on this distribution, indicated by j_t. The adversary then chooses a loss vector, $\boldsymbol{\ell}_t$, such that for all j, $\boldsymbol{\ell}_t(j) \in [-\kappa, \kappa]$. The adversary is adaptive, in the sense that the choice of $\boldsymbol{\ell}_t$ can depend on the distributions at every step $\mathbf{q}_1, \ldots, \mathbf{q}_t$, and on the realized actions in the previous steps $j_1, \ldots, j_{t-1}$. The online learner then incurs an expected loss of $\mathbf{q}_t \cdot \boldsymbol{\ell}_t$.

Let $L_{alg} = \sum_{t=1}^{T} \mathbf{q}_t \cdot \boldsymbol{\ell}_t$ be the total expected loss of the online learner over time period T for a choice of an algorithm. On the other hand, let $L_{\min} = \min_{j \in \mathcal{M}} \sum_{t=1}^{T} \boldsymbol{\ell}_t(j)$, be the loss of the best fixed action for the sequence $\boldsymbol{\ell}_1, \ldots, \boldsymbol{\ell}_T$. Define *regret* as $R_{T,\mathcal{M},\kappa} = L_{alg} - L_{min}$.

In our work we leverage a well-known result on no-regret learning, which can be stated as follows (see, e.g., [Blum and Mansour 2007]).

PROPOSITION 1. *There is an algorithm such that*

$$R_{T,\mathcal{M},\kappa} \leq \sqrt{T\kappa \log(|\mathcal{M}|)}$$

In this work, we can use any algorithm that satisfies the above guarantee as a black box. Many algorithms fall into this category, e.g., *Polynomial Weights* [Cesa-Bianchi et al. 2007] and *Follow the Lazy Leader* [Kalai and Vempala 2005]. In section 8, we discuss the choice of algorithm further. Also note that any utility maximization problem has an equivalent loss minimization formulation. To be consistent with the notation used by the learning community, we adopt the notion of loss minimization when dealing with results pertaining to online learning.

Although our problem is closely related to classic regret minimization, our goal cannot be readily accomplished by applying Proposition 1. Indeed, each *mixed* strategy in a security game corresponds to one action in Proposition 1. This creates an infinitely large set of actions, which renders the guarantees given by Proposition 1 meaningless. Previous work resolves this issue for a subset of problems where the action space is itself a vector space and the loss function has some desirable properties, e.g., linear or convex with some restrictions [Bubeck and Cesa-Bianchi 2012; Kalai and Vempala 2005; Zinkevich 2003]. However, the loss function used in our work does not have such nice structural properties: the loss function at step t depends on the best response of the attacker, which leads to a loss function that is not linear, convex, or even continuous in the mixed strategy.

4. CHARACTERISTICS OF THE OFFLINE OPTIMUM

In this section, we examine the characteristics of the best-in-hindsight mixed strategy. Using this characterization, we choose a set of mixed strategies that are representative of the continuous space of all mixed strategies. That is, rather than considering all mixed strategies in $\mathcal{P}$, we show that we can limit the choices of our online algorithm to a subset of mixed strategies without incurring (significant) additional regret.

To this end, we first show that the given attacker types partition the space of mixed strategies into convex regions where the attacker's best response remains fixed.

Definition 4.1. For every target $i \in N$ and attacker type $\alpha_j \in K$, let $\mathcal{P}_i^j$ indicate the set of all valid coverage probabilities where an attacker of type α_j attacks target i, i.e.,

$$\mathcal{P}_i^j = \{\mathbf{p} \in \mathcal{P} \mid b_{\alpha_j}(\mathbf{p}) = i\}.$$

It is known that for all i and j, $\mathcal{P}_i^j$ is a convex polytope [Blum et al. 2014b].

Definition 4.2. For a given function $\sigma : K \to N$, let $\mathcal{P}_\sigma$ indicate the set of all valid coverage probability vectors such that for all $\alpha_j \in K$, α_j attacks $\sigma(j)$. In other words, $\mathcal{P}_\sigma = \bigcap_{j \in K} \mathcal{P}_{\sigma(j)}^j$.

Note that $\mathcal{P}_\sigma$ is an intersection of finitely many convex polytopes, so it is itself a convex polytope. Let Σ indicate the set of all σ for which $\mathcal{P}_\sigma$ is a non-empty region. Figure 1 illustrates these regions.

The next lemma essentially shows that the optimal strategy in hindsight is an extreme point of one of the convex polytopes defined above. There is one subtlety, though: due to tie breaking, for some σ, $\mathcal{P}_\sigma$ is not necessarily closed. Hence, some of the extreme points of the closure of $\mathcal{P}_\sigma$ are not within that region. To circumvent this issue, instead we prove that the optimal strategy in hindsight is *approximately* an extreme point of one of the regions. That is, for a given $\epsilon > 0$, we define $\mathcal{E}$ to be a set of mixed strategies as follow: for all σ and any $\mathbf{p}$ that is an extreme point of the closure of $\mathcal{P}_\sigma$, if $\mathbf{p} \in \mathcal{P}_\sigma$, then $\mathbf{p} \in \mathcal{E}$, otherwise there exists $\mathbf{p}' \in \mathcal{E}$ such that $\mathbf{p}' \in \mathcal{P}_\sigma$ and $||\mathbf{p} - \mathbf{p}'||_1 \leq \epsilon$.

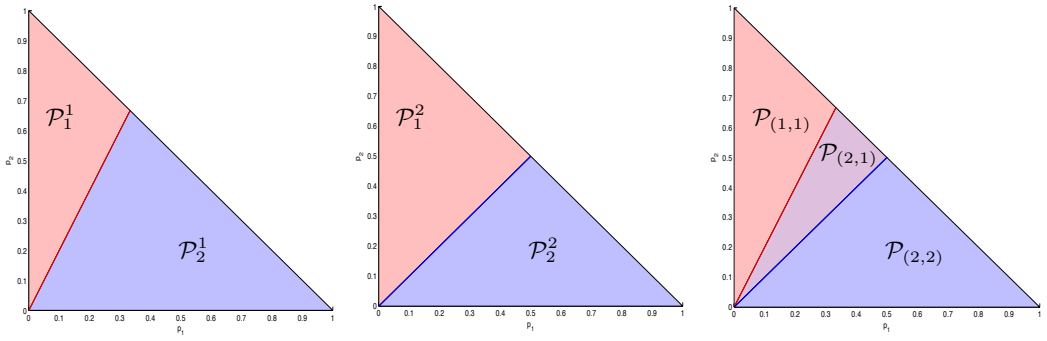

Fig. 1: Best-response regions. The first two figures define $\mathcal{P}_i^j$ in a game where one resource can cover one of two targets, and two attacker types. The third figure illustrates $\mathcal{P}_\sigma$ for the intersection of the best-response regions of the two attackers.

LEMMA 4.3. *Let $\mathcal{E}$ be as defined above, then for any sequence of attackers $\mathbf{a}$,*

$$\max_{\mathbf{p}\in\mathcal{E}} \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}), \mathbf{p}) \geq \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - 2\epsilon T.$$

PROOF. Recall that $\mathbf{p}^*$ is the optimal strategy in hindsight for a sequence of attackers $\mathbf{a}$. Since the set of regions $\mathcal{P}_\sigma$ for all $\sigma : K \to N$ partitions the set of all mixed strategies, $\mathbf{p}^* \in \mathcal{P}_\sigma$ for some σ. We will show that there is a point $\mathbf{p}'' \in \mathcal{E}$ that is near optimal for the sequence $\mathbf{a}$.

For any coverage probability $\mathbf{p} \in \mathcal{P}_\sigma$, let $\mathcal{U}_\mathbf{a}(\mathbf{p})$ be the defender's expected payoff for playing $\mathbf{p}$ against sequence $\mathbf{a}$. Then,

$$\mathcal{U}_\mathbf{a}(\mathbf{p}) = \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}), \mathbf{p}) = \sum_{t=1}^{T} U_d(\sigma(a_t), \mathbf{p}) = \sum_{i=1}^{N} U_d(i, \mathbf{p}) \sum_{t=1}^{T} \mathbb{I}(\sigma(a_t) = i),$$

where $\mathbb{I}$ is the indicator function. Note that $\sum_{t=1}^{T} \mathbb{I}(\sigma(a_t) = i)$, which is the total number of times target i is attacked in the sequence $\mathbf{a}$, is constant over $\mathcal{P}_\sigma$. Moreover, by the definition of utilities, $U_d(i, \mathbf{p})$ is a linear function in $\mathbf{p}$. Therefore, $\mathcal{U}_\mathbf{a}(\mathbf{p})$ is a summation of linear functions, and as a result, is itself a linear function in $\mathbf{p}$ over $\mathcal{P}_\sigma$.

Let $\mathcal{P}'_\sigma$ be the closure of $\mathcal{P}_\sigma$. So, $\mathcal{P}'_\sigma$ is a closed convex polytope. Since $\mathcal{U}_\mathbf{a}(\mathbf{p})$ is a linear function over the convex set $\mathcal{P}'_\sigma$, there is an extreme point $\mathbf{p}' \in \mathcal{P}'_\sigma$ such that

$$\sum_{t=1}^{T} U_d(\sigma(a_t), \mathbf{p}') \geq \mathcal{U}_\mathbf{a}(\mathbf{p}^*).$$

Now, let $\mathbf{p}'' \in \mathcal{E} \cap \mathcal{P}_\sigma$ be the point that corresponds to extreme point $\mathbf{p}'$, i.e. $||\mathbf{p}' - \mathbf{p}''||_1 \leq \epsilon$. Since for each target $i \in N$, $u_d^c(i), u_d^u(i) \in [-1, 1]$ we have

$$U_d(i, p_i'') = u_d^c(i)p_i'' + u_d^u(i)(1 - p_i'') \geq u_d^c(i)(p_i'' + \epsilon) + u_d^u(i)(1 - p_i'' - \epsilon) - 2\epsilon \geq U_d(i, p_i') - 2\epsilon.$$

Hence,

$$\mathcal{U}_\mathbf{a}(\mathbf{p}'') = \sum_{t=1}^{T} U_d(\sigma(a_t), \mathbf{p}'') \geq \sum_{t=1}^{T} U_d(\sigma(a_t), \mathbf{p}') - 2\epsilon T \geq \mathcal{U}_\mathbf{a}(\mathbf{p}^*) - 2\epsilon T.$$

□

The above lemma shows that by only considering strategies in $\mathcal{E}$, our online algorithm will merely incur an additional loss of ϵT. For a small enough choice of ϵ this only adds a lower-order term to our regret analysis, and as a result can effectively be ignored. For the remainder of this paper, we assume that $\mathcal{E}$ is constructed with $\epsilon \in O(\frac{1}{\sqrt{t}})$. Next, we derive an upper bound on the size of $\mathcal{E}$.

LEMMA 4.4. *For any repeated security game with n targets and k attacker types, $|\mathcal{E}| \in O((2^n + kn^2)^{2n})$.*

PROOF. Any extreme point of a convex polytope in n dimensions is the intersection of n linearly independent defining half-spaces of that polytope. To compute the number of extreme points, we first compute the number of defining half-spaces. For a given attacker type α_j, there are $\binom{n}{2}$ half-spaces each separating $\mathcal{P}_i^j$ and $\mathcal{P}_{i'}^j$ for two targets $i \neq i'$. Summing over all attacker types, there are $O(kn^2)$ such half-spaces. Moreover, the region of the valid coverage probabilities is itself the intersection of $O(m+n)$ half-spaces, where m is the number of subsets of targets that are covered in some pure strategy [Blum et al. 2014b]. In the worst case, $m = 2^n$. Therefore, each extreme point is an intersection of n halfspaces chosen from $O(2^n + n + kn^2)$ half-spaces, resulting in at most $\binom{m+n+kn^2}{n} \in O((2^n + kn^2)^n)$ extreme points. Each extreme point can give rise to at most n^k other ϵ-approximate extreme points. Therefore, $|\mathcal{E}| \in O((2^n + kn^2)^n n^k)$. □

5. UPPER BOUNDS – FULL INFORMATION

In this section, we establish an upper bound on regret in the full information setting. With the machinery of Section 4 in place, the proof follows quite easily from Proposition 1.

THEOREM 5.1. *Given a repeated security game with full information feedback, n targets, k attacker types, and time horizon T[1], there is an online algorithm that for any unknown sequence of attackers, $\mathbf{a}$, at time t plays a randomly chosen mixed strategy $\mathbf{p}_t$, and has the property that*

$$\mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right] \geq \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - O\left(\sqrt{Tn^2 k \log nk}\right),$$

where the expectation is taken over the algorithm's internal randomization.

Before proving the theorem, a comment is in order. The algorithm is playing a distribution over mixed strategies at any stage; isn't that just a mixed strategy? The answer is no: the attacker is best-responding to the mixed strategy drawn from the distribution over mixed strategies chosen by the defender. The best responses of the attacker could be completely different if he responded directly to the mixed strategy induced by this distribution over mixed strategies.

PROOF OF THEOREM 5.1. We use any algorithm that satisfies Proposition 1. Let the set of extreme points $\mathcal{E}$ denote the set of actions to be used in conjunction with this algorithm. At every round, after observing a_t, we compute the loss of all mixed strategies $\mathbf{p} \in \mathcal{E}$ by setting $\ell_t(\mathbf{p}) = -U_d(b_{a_t}(\mathbf{p}), \mathbf{p})$. Note that the problem of maximizing utility is now transformed to mini-

[1]If T is unknown, we can use the *guess and double* technique, adding to the regret bound an extra constant factor.

mizing the loss. Using Proposition 1 and Lemma 4.4, we have

$$\begin{aligned}\mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right] &\geq \max_{\mathbf{p}\in\mathcal{E}} \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}), \mathbf{p}) - O(\sqrt{T \log |\mathcal{E}|}) \\ &\geq \max_{\mathbf{p}\in\mathcal{E}} \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}), \mathbf{p}) - O\left(\sqrt{T(n\log(2^n + kn^2) + k\log n)}\right) \\ &\geq \max_{\mathbf{p}\in\mathcal{E}} \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}), \mathbf{p}) - O\left(\sqrt{Tn^2 k \log nk}\right).\end{aligned}$$

Using Lemma 4.3 with an appropriate choice of $\epsilon \in O(\frac{1}{\sqrt{T}})$, we have

$$\mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right] \geq \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - O\left(\sqrt{Tn^2 k \log nk}\right).$$

□

6. UPPER BOUNDS – PARTIAL INFORMATION

In this section, we establish an upper bound on regret in the partial information setting. Recall that under partial information feedback, after an attack has occurred, the defender only observes the target that was attacked, not the attacker type that initiated the attack. Our goal is to prove the following theorem.

THEOREM 6.1. *Given a repeated security game with partial information feedback, n targets, k attacker types, and time horizon T, there is an online algorithm that for any unknown sequence of attackers, $\mathbf{a}$, at time t plays a randomly chosen mixed strategy $\mathbf{p}_t$, and has the property that*

$$\mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right] \geq \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - O\left(T^{2/3}\, nk\, \log^{1/3}(nk)\right),$$

where the expectation is taken over algorithm's internal randomization.

6.1. Overview of the Approach

The central idea behind our approach is that any regret-minimization algorithm in the full information setting also works with partial feedback if, instead of observing the loss of every action at a time step, it receives an *unbiased estimator* of the loss of all actions. For time horizon T and range of action loss $[-\kappa, +\kappa]$, let $R_{T,\kappa}$ represent the loss of a full-information algorithm. For any action j and any time step t, let $\hat{\ell}_t(j)$ be an unbiased estimator of the loss of action j at time step t. Run the full-information regret-minimization algorithm with the values of the loss estimators, i.e. $\hat{\ell}_t(j)$, and let $q_t(j)$ be the probability that our full-information algorithm picks action j at time t. Then, the expected loss of this algorithm (denoted $L_{Est.}$) is as follows.

$$\begin{aligned}L_{Est.} &= \sum_{t=1}^{T}\sum_{j\in\mathcal{M}} q_t(j)\ell_t(j) = \sum_{t=1}^{T}\sum_{j\in\mathcal{M}} q_t(j)\mathbb{E}[\hat{\ell}_t(j)] = \mathbb{E}\left[\sum_{t=1}^{T}\sum_{j\in\mathcal{M}} q_t(j)\hat{\ell}_t(j)\right] \\ &\leq \mathbb{E}\left[\min_j \sum_{j\in\mathcal{M}} \hat{\ell}_t(j) + R_{T,\kappa}\right] \leq \min_j \mathbb{E}\left[\sum_{j\in\mathcal{M}} \hat{\ell}_t(j)\right] + R_{T,\kappa} = \min_j \sum_{j\in\mathcal{M}} \ell_t(j) + R_{T,\kappa}.\end{aligned} \tag{1}$$

This means that, in the partial information setting, we can use a full-information regret-minimization algorithm in combination with a mechanism to estimate the loss of all actions. This is similar to the approach first introduced by Awerbuch and Mansour [2003]. Informally speaking, we simulate one time step of a full information setting by a window of time, where we mostly use the mixed strategy that is suggested by the full information algorithm (exploitation), except for a few time steps where we invoke a mechanism for getting an estimate for the loss of all mixed strategies in that window of time (exploration). We then pass these estimates to the full-information algorithm to be used for future time steps.

A naïve approach for getting unbiased estimators of the loss of all mixed strategies involves sampling each mixed strategy once at random in a window of time and observing its loss. Note that, at every time step in which we sample a random strategy, we could incur significant regret. That is, the strategy that is being sampled may have significant loss compared to the strategy suggested by the full-information regret minimization algorithm. Therefore, sampling each strategy individually once at random adds regret that is polynomial in the number of mixed strategies sampled, which in our case is *exponential* in the number of targets and types (See Lemma 6.2 for more details). Therefore, a more refined mechanism for finding an unbiased estimator of the loss is needed. That is, the question is: *How can we get an unbiased estimator of the loss of all mixed strategies while sampling only a few of them?*

To answer this question, we first notice that the loss of different mixed strategies is not independent, rather they all depend on the number of times each target is attacked, which in turn depends on the type frequency of attackers. The challenge, then, is to infer an unbiased estimator of the type frequencies, by only observing the best responses (not the types themselves).

As an example, assume that there is a mixed strategy where each attacker type responds differently (See Figure 1). Then by observing the best-response, we can infer the type with certainty. In general, such a mixed strategy, where each attacker responds differently, might not exists. This is where insights from *bandit linear optimization* prove useful.

In more detail, in order to estimate the loss of a mixed strategy $\mathbf{p} \in \mathcal{P}_\sigma$, it is sufficient to estimate the total frequency of the set of attacker types that attack target i in region $\mathcal{P}_\sigma$, for any $i \in N$. This value itself is *a linear function* in $\mathbb{R}^k$. That is, let $\mathbb{I}_{(\sigma=i)}$ be the indicator vector of the set of attackers that attack i in region $\mathcal{P}_\sigma$ and let $\mathbf{f} = (f_1, \ldots, f_k) \in \mathbb{R}^k$ be the vector of frequencies of attacker types. The loss of mixed strategy $\mathbf{p}$ can be determined using values $\mathbf{f} \cdot \mathbb{I}_{(\sigma=i)}$ for all $i \in N$. Moreover, even though we cannot observe the above inner products directly under partial information feedback, we can create an unbiased estimator of any $\mathbf{f} \cdot \mathbb{I}_{(\sigma=i)}$ by sampling any $\mathbf{p} \in \mathcal{P}_\sigma$ and observing how often target i is attacked in response. Therefore our problem reduces to creating "good" unbiased estimators for the above inner products, which lie in a k-dimensional vector space. This can be achieved by only sampling a set of k strategies (See Lemmas 6.3 and 6.4)

One subtle obstacle remains, and that is the range of the new loss estimator (where the notion of a "good" estimator comes in). Indeed, as shown in Eq. 1, the regret also depends on the range of the loss estimator. To this end, we use the concept of *barycentric spanners* [Awerbuch and Kleinberg 2008] that is also used in bandit linear optimization, to ensure that the range of the loss estimator remains small even after inferring it through frequency estimation.

6.2. Partial Information to Full Information

As briefly discussed earlier, any regret-minimization problem with partial-information feedback can be reduced to the full information case, assuming that we can estimate the loss of all actions by sampling a subset of the actions. The next lemma gives an upper bound on regret in terms of the number of actions sampled, the quality of the estimator (in terms of its range), and the total number of actions.

LEMMA 6.2. *Let $\mathcal{M}$ be the set of all actions. For any time block (set of consecutive time steps) T' and action $j \in \mathcal{M}$, let $c_{T'}(j)$ be the* average loss *of action j over T'. Assume that $\mathcal{S} \subseteq \mathcal{M}$ is such that by sampling all actions in $\mathcal{S}$, we can compute $\hat{c}_{T'}(j)$ for all $j \in \mathcal{M}$ with the following*

properties:

$$\mathbb{E}[\hat{c}_{T'}(j)] = c_{T'}(j) \quad \textit{and} \quad \hat{c}_{T'}(j) \in [-\kappa, \kappa].$$

Then there is an algorithm with loss

$$L_{alg} \leq L_{\min} + O\left(T^{2/3}|\mathcal{S}|^{1/3}\kappa^{1/3}\log^{1/3}(|\mathcal{M}|)\right),$$

where $L_{\min}$ is the loss of the best action in hindsight.

PROOF. Our proposed algorithm is as follows. Let $Z = \left(T^2|\mathcal{S}|^{-2}\kappa\log(|\mathcal{M}|)\right)^{1/3}$. Divide T into Z (roughly) equal blocks, $B_1, \ldots, B_Z$. In each block, pick a uniformly random permutation of $\mathcal{S}$ together with $|\mathcal{S}|$ uniformly random time steps in that block, and assign them to the *exploration phase*. At any time step that is dedicated to exploration, sample the actions in $\mathcal{S}$ in the order they appear in the random permutation. At the end of each block, by the assumptions of the lemma, we receive $\hat{c}_\tau(j)$, which is the average loss of action j during B_τ. We pass this loss information to any regret-minimization algorithm that works in the full information feedback model. At every time step that is not designated for exploration, we use the action that is computed by the full-information algorithm based on the loss from the previous time block.

Next, we compute the regret of the proposed algorithm: Let $q_\tau(\cdot)$ be the fixed probability distribution over actions suggested by the full information algorithm for block B_τ. On the other hand, let $q_t(\cdot)$ be the probability distribution over actions used by the partial information algorithm. That is, during exploitation time steps t in block B_τ, $q_t(\cdot) = q_\tau(\cdot)$, and during the exploration phase $q_t(\cdot)$ refers the action that is being sampled at round t. For any action j, let $\ell_t(j)$ represent the actual loss at round t, and $L_\tau(j)$ and $c_\tau(j)$ indicate the aggregate and average loss of j in block B_τ, respectively. That is

$$L_\tau = \sum_{t \in B_\tau} \ell_t(j) \quad \text{and} \quad c_\tau = \frac{L_\tau(j)}{|B_\tau|}.$$

We have

$$\begin{aligned}
L_{alg} &= \sum_{\tau=1}^{Z} \sum_{t \in B_\tau} \sum_{j \in \mathcal{M}} q_t(j)\ell_t(j) \\
&\leq \sum_{\tau=1}^{Z} \sum_{j \in \mathcal{M}} q_\tau(j) L_\tau(j) + Z \cdot |\mathcal{S}| \quad \text{(Each } j \in \mathcal{S} \text{ is sampled once and has loss} \leq 1) \\
&\leq \sum_{\tau=1}^{Z} \sum_{j \in \mathcal{M}} q_\tau(j)\mathbb{E}[\hat{c}_\tau(j)] \left(\frac{T}{Z}\right) + Z \cdot |\mathcal{S}| \quad \text{(Definition of } c_\tau(\cdot) \text{ and unbiasedness of } \hat{c}_\tau(\cdot)) \\
&\leq \frac{T}{Z}\, \mathbb{E}\left[\sum_{\tau=1}^{Z} \sum_{j \in \mathcal{M}} q_\tau(j)\hat{c}_\tau(j)\right] + Z \cdot |\mathcal{S}| \\
&\leq \frac{T}{Z}\, \mathbb{E}\left[\min_j \sum_{\tau=1}^{Z} \hat{c}_\tau(j) + R_{Z,\kappa}\right] + Z \cdot |\mathcal{S}| \quad \text{(Regret bound under full information)} \\
&\leq \frac{T}{Z}\, \left(\min_j \mathbb{E}\left[\sum_{\tau=1}^{Z} \hat{c}_\tau(j)\right] + R_{Z,\kappa}\right) + Z \cdot |\mathcal{S}| \quad \text{(Jensen's Inequality)} \\
&\leq \frac{T}{Z}\, \left(\min_j \sum_{\tau=1}^{Z} c_\tau(j) + R_{Z,\kappa}\right) + Z \cdot |\mathcal{S}| \quad \text{(Unbiasedness of } \hat{c}_\tau(j))
\end{aligned}$$

$$\leq \min_j \sum_{t=1}^{T} \ell_t(j) + \frac{T}{Z} R_{Z,\kappa} + Z \cdot |\mathcal{S}| \quad \text{(Because } \sum_{t=1}^{T} \ell_t(j) = \sum_{\tau=1}^{Z} c_\tau(j)|B_\tau|\text{)}$$

$$\leq L_{\min}^T + \frac{T}{Z}\sqrt{Z\kappa \log(|\mathcal{M}|)} + Z \cdot |\mathcal{S}|$$

$$\leq L_{\min}^T + O\left(T^{2/3}|\mathcal{S}|^{1/3}\kappa^{1/3}\log^{1/3}(|\mathcal{M}|)\right) \quad \text{(Because } Z = \left(T^2|\mathcal{S}|^{-2}\kappa \log(|\mathcal{M}|\right)^{1/3}\text{)}$$

□

6.3. Creating Unbiased Estimators

Constructing an unbiased loss estimator for each mixed strategy is a pre-requisite for our reduction to the full information case (as in Lemma 6.2). Here, we show how such estimators can be constructed for all mixed strategies, by sampling only a small number of mixed strategies.

For any τ, let $f_\tau : \mathbb{R}^k \to \mathbb{R}$ be a function that for any $\mathbf{w} = (w_1, \ldots, w_k)$ returns the number of times attacker types $\alpha_1, \ldots, \alpha_k$ were active in block B_τ, weighted by coefficients of $\mathbf{w}$. That is

$$f_\tau(\mathbf{w}) = \sum_{i=1}^{k} w_j \sum_{t \in B_\tau} \mathbb{I}_{(a_t = \alpha_j)},$$

where $\mathbb{I}$ is an indicator function. Note that for $K' \subseteq K$ and its corresponding indicator vector $\mathbf{w}$, $f(\mathbf{w})$ is the total number times attackers in K' are active in B_τ. Furthermore, note that $\sum_{t \in B_\tau} \mathbb{I}_{(a_t=\alpha_j)}$ is a constant for any τ and j, therefore $f_\tau(\cdot)$ is a linear function.

For any mixed strategy $\mathbf{p} \in P_\sigma$ and any τ, let $c_\tau(\mathbf{p})$ represent the *average* utility of $\mathbf{p}$ against attackers in block B_τ. Then,

$$c_\tau(\mathbf{p}) = \frac{1}{|B_\tau|} \sum_{i=1}^{N} U_d(i, \mathbf{p}) f_\tau(\mathbb{I}_{(\sigma=i)}),$$

where $\mathbb{I}_{(\sigma=i)}$ is an indicator vector representing all the attackers that respond to a mixed strategy in region $\mathcal{P}_\sigma$ (including $\mathbf{p}$) by attacking i.

Our goal is to construct an unbiased estimator for $c_\tau(\mathbf{p})$ for any $\mathbf{p}$. To do so, we first construct an estimator for $f_\tau(\cdot)$. Since $f_\tau(\cdot)$ is linear in $\mathbb{R}^k$, it suffices to construct an estimator for a spanning set of $\mathbb{R}^k$. To this end, we define $\mathcal{W} = \{\mathbb{I}_{(\sigma=i)}| \text{ for all } i \in N \text{ and } \sigma \in \Sigma\}$, which is a set of vectors $\mathbb{I}_{(\sigma=i)}$ for any i and σ, each corresponding to attacker types that attack target i in region $\mathcal{P}_\sigma$. Next, we introduce a result by Awerbuch and Kleinberg [2008] that helps us in choosing an appropriate basis for $\mathcal{W}$.

PROPOSITION 2. *([Awerbuch and Kleinberg 2008, Proposition 2.2]) If $\mathcal{W}$ is a compact subset of d-dimensional vector space $\mathcal{V}$, then there exists a set $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_d\} \subseteq \mathcal{W}$ such that for all $\mathbf{w} \in \mathcal{W}$, $\mathbf{w}$ may be expressed as a linear combination of elements of $\mathcal{B}$ using coefficients in $[-1, 1]$. That is, for all $\mathbf{w} \in \mathcal{W}$, there exist coefficients $\lambda_1, \ldots, \lambda_d \in [-1, 1]$, such that $\mathbf{w} = \sum \lambda_j \mathbf{b}_j$. Such $\mathcal{B}$ is called a* barycentric spanner *for $\mathcal{W}$.*

Consider the set $\mathcal{W}$. First, note that $\mathcal{W}$ is finite, so it is compact and it has a barycentric spanner of size k. Using the methods of Awerbuch and Kleinberg [2008], we can construct a barycentric spanner for $\mathcal{W}$ by performing linear optimization over $\mathcal{W}$. Alternatively, for a choice of $\{\mathbf{b}_1, \ldots, \mathbf{b}_k\} \in \mathcal{W}$, and for all $\mathbf{w} \in \mathcal{W}$, we can solve the following feasibility LP:

$$\forall j \in [k], \ \sum_i \lambda_i b_{i,j} = w_j,$$

$$\forall i, \ -1 \leq \lambda_i \leq +1.$$

Let $\mathcal{B}$ be the barycentric spanner for $\mathcal{W}$ as defined above. For any $\mathbf{b} \in \mathcal{B}$, there must be a mixed strategy $\mathbf{p} \in \mathcal{P}_\sigma$ and target $i \in N$ such that $\mathbb{I}_{(\sigma=i)} = \mathbf{b}$ (otherwise $\mathbf{b} \notin \mathcal{W}$); call such strategy and target $\mathbf{p_b}$ and $i_\mathbf{b}$, respectively. We use $\mathbf{p_b}$ and $i_\mathbf{b}$ for the purpose of creating a loss estimator for $f_\tau(\mathbf{b})$ as follows: In the exploration phase, once at random (based on a chosen random permutation), we play $\mathbf{p_b}$ and observe target $i_\mathbf{b}$. If $i_\mathbf{b}$ is attacked in response we set $\hat{p}_\tau(\mathbf{b}) = 1$, otherwise $\hat{p}_\tau(\mathbf{b}) = 0$. The next lemma shows that $\hat{p}_\tau(\mathbf{b})|B_\tau|$ is an unbiased estimator for $f_\tau(\mathbf{b})$.

LEMMA 6.3. *For any* $\mathbf{b} \in \mathcal{B}$, $\mathbb{E}\left[\hat{p}_\tau(\mathbf{b}) \cdot |B_\tau|\right] = f_\tau(\mathbf{b})$.

PROOF. Note that $\hat{p}(\mathbf{b}) = 1$ if and only if at the time step that $\mathbf{p_b}$ was played for the purpose of recording $\mathbf{b}$, target $i_\mathbf{b}$ was attacked. Because $\mathbf{p_b}$ is played once for the purpose of recording $\mathbf{b}$ uniformly at random over the time steps and the adversarial sequence is chosen before the game play, the attacker that responded to $\mathbf{p_b}$ is also picked uniformly at random over the time steps. Therefore, $\mathbb{E}[\hat{p}(\mathbf{b})]$ is the probability that a randomly chosen attacker from B_τ responds in a way that is consistent with any one of the attackers who responds by attacking $i_\mathbf{b}$. Formally,

$$\mathbb{E}[\hat{p}_\tau(\mathbf{b})] = \frac{\sum_{i:b_i=1} f_\tau(\mathbf{e}_i)}{|B_\tau|} = \frac{f_\tau(\mathbf{b})}{|B_\tau|}.$$

□

Since $\mathcal{W} \subseteq \{0,1\}^k$, the rank of $\mathcal{W}$ is at most k. Let $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ be the barycentric spanner for $\mathcal{W}$. For any $\mathbf{w} \in \mathcal{W}$, let $\boldsymbol{\lambda}(\mathbf{w})$ be the representation of $\mathbf{w}$ in basis $\mathcal{B}$. That is, $\sum_{j=1}^k \lambda_j(\mathbf{w})\, \mathbf{b}_j = \mathbf{w}$. Now, consider any mixed strategy $\mathbf{p}$ and let σ be such that $\mathbf{p} \in \mathcal{P}_\sigma$. Let

$$\hat{c}_\tau(\mathbf{p}) = \sum_{i=1}^{n} \sum_{j=1}^{k} \lambda_j(\mathbb{I}_{\sigma=i})\, \hat{p}_\tau(\mathbf{b}_j) U_d(i, \mathbf{p}). \tag{2}$$

The next lemma shows that for all $\mathbf{p}$, $\hat{c}_\tau(\mathbf{p})$ is indeed an unbiased estimator of $c_\tau(\mathbf{p})$ with a small range.

LEMMA 6.4. *For any mixed strategy* $\mathbf{p}$, $\mathbb{E}[\hat{c}_\tau(\mathbf{p})] = c_\tau(\mathbf{p})$ *and* $\hat{c}_\tau(\mathbf{p}) \in [-nk, nk]$.

PROOF. Let σ be such that $\mathbf{p} \in \mathcal{P}_\sigma$. We have,

$$\begin{aligned}
\mathbb{E}[\hat{c}_\tau(\mathbf{p})] &= \mathbb{E}\left[\sum_{i=1}^{n}\sum_{j=1}^{k} \lambda_j(\mathbb{I}_{\sigma=i})\, \hat{p}_\tau(\mathbf{b}_j) U_d(i, \mathbf{p})\right] \\
&= \sum_{i=1}^{n}\sum_{j=1}^{k} \lambda_j(\mathbb{I}_{\sigma=i})\, \mathbb{E}[\hat{p}_\tau(\mathbf{b}_j)]\, U_d(i, \mathbf{p}) && \text{(linearity of expectation)} \\
&= \sum_{i=1}^{n} U_d(i, \mathbf{p}) \sum_{j=1}^{k} \frac{\lambda_j(\mathbb{I}_{\sigma=i})\, f_\tau(\mathbf{b}_j)}{|B_\tau|} && \text{(by Lemma 6.3)} \\
&= \sum_{i=1}^{n} \frac{U_d(i, \mathbf{p})}{|B_\tau|} f_\tau\left(\sum_{j=1}^{k} \lambda_j(\mathbb{I}_{\sigma=i})\, \mathbf{b}_j\right) && \text{(by linearity of } f_\tau) \\
&= \sum_{i=1}^{n} \frac{U_d(i, \mathbf{p})}{|B_\tau|} f_\tau(\mathbb{I}_{(\sigma=i)}) && \text{(by definition of } \boldsymbol{\lambda}(\cdot)) \\
&= c_\tau(\mathbf{p}).
\end{aligned}$$

Since $\mathcal{B}$ is barycentric, for any $\mathbf{w} \in \mathcal{W}$, $\lambda_j(\mathbf{w}) \in [-1, 1]$. Moreover, $\hat{p}_\tau(\cdot) \in \{0, 1\}$, and $U_d(i, \mathbf{p}) \in [-1, 1]$. So,

$$\hat{c}_\tau(\mathbf{p}) = \sum_{i=1}^{n} \sum_{j=1}^{k} \lambda_j(\mathbb{I}_{\sigma=i})\, \hat{p}_\tau(\mathbf{b}_j) U_d(i, \mathbf{p}) \in [-nk, nk].$$

□

6.4. Putting It All Together

Using the machinery developed in previous subsections, we can now proceed to prove the theorem.

PROOF OF THEOREM 6.1. We use Algorithm 1 along with Proposition 1 as a black-box full-information regret minimization algorithm. We use $\mathcal{E}$ as the set of mixed strategies described in Section 4.

ALGORITHM 1: REPEATED SECURITY GAMES WITH PARTIAL INFORMATION

Oracle: Black-box access to an algorithm that satisfies Proposition 1, FULL-INFORMATION$(\cdot)$, which takes as input the loss of all actions and produces a distribution $\mathbf{q}$ over them.
Algorithm:

(1) $Z \leftarrow \left(T^2 n^2 k \log nk\right)^{1/3}$
(2) Create set $\mathcal{W} = \{\mathbb{I}_{(\sigma=i)} | \text{ for all } i \in N \text{ and } \sigma \in \Sigma\}$.
(3) Find a barycenteric spanner $\mathcal{B} = \{\mathbf{b}_1, \ldots, \mathbf{b}_k\}$ for $\mathcal{W}$. For every $\mathbf{b} \in \mathcal{B}$ such that $\mathbf{b} = \mathbb{I}_{(\sigma=i)}$ for some i and σ, let $i_{\mathbf{b}} \leftarrow i$ and $\mathbf{p}_{\mathbf{b}}$ be any mixed strategy in $\mathcal{P}_\sigma$.
(4) For all $\mathbf{w} \in \mathcal{W}$ let $\boldsymbol{\lambda}(\mathbf{w})$ be the representation of $\mathbf{w}$ in basis $\mathcal{B}$. That is $\sum_{j=1}^{k} \lambda_j(\mathbf{w})\, \mathbf{b}_j = \mathbf{w}$.
(5) Let $\mathbf{q}_1$ be the uniform distribution over $\mathcal{E}$.
(6) For $\tau = 1, \ldots, Z$
  (a) Choose a random permutation π over $[k]$ and $t_1, \ldots, t_k$ time steps at random from $[T/Z]$.
  (b) For $t = (\tau - 1)(T/Z) + 1, \ldots, \tau(T/Z)$,
    i. If $t = t_j$ for some $j \in [k]$, then $\mathbf{p}_t \leftarrow \mathbf{p}_{\mathbf{b}_{\pi(j)}}$. If $i_{\mathbf{b}_{\pi(j)}}$ is attacked, then $\hat{p}_\tau(\mathbf{b}_{\pi(j)}) \leftarrow 1$, otherwise $\hat{p}_\tau(\mathbf{b}_{\pi(j)}) \leftarrow 0$.
    ii. Else,
    draw $\mathbf{p}_t$ at random from distribution $\mathbf{q}_\tau$.
  (c) For all $\mathbf{p} \in \mathcal{E}$, for σ such that $\mathbf{p} \in \mathcal{P}_\sigma$,

$$\hat{c}_\tau(\mathbf{p}) \leftarrow \sum_{i=1}^{n} \sum_{j=1}^{k} \lambda_j(\mathbb{I}_{\sigma=i})\, \hat{p}_\tau(\mathbf{b}_j) U_d(i, \mathbf{p}).$$

  (d) Call FULL-INFORMATION$(\hat{c}_\tau)$. And receive $\mathbf{q}_{\tau+1}$ as a distribution over all mixed strategies in $\mathcal{E}$.

Our algorithm divides the timeline to $Z = \left(T^2 n^2 k \log nk\right)^{1/3}$ equal intervals, $B_1, \ldots, B_Z$. The initial distribution over the set of mixed strategies $\mathcal{E}$ is the uniform distribution.

In each block, we pick a random permutation π over $\mathcal{B}$ together with k time steps in that block and mark them for exploration. At the j^{th} time step that is dedicated to exploration, we play mixed strategy $\mathbf{p}_{b_{\pi(j)}}$ and observe target $i_{b_{\pi(j)}}$. If $i_{b_{\pi(j)}}$ is attacked, then we assign $\hat{p}_\tau(b_{\pi(j)}) = 1$ otherwise $\hat{p}_\tau(b_{\pi(j)}) = 0$. At any time step that is not set for exploration, we choose a mixed strategy at random from the current distribution over $\mathcal{E}$.

At the end of each block, we compute $\hat{c}_\tau(\mathbf{p})$ for all $\mathbf{p} \in \mathcal{E}$, using Equation 2. We then pass this loss information to any regret-minimization algorithm that works in the full information feedback model, and update the default distribution based on its outcome.

Using Lemma 6.4, $\hat{c}_\tau(\mathbf{p})$ is an unbiased estimator of the average loss of action $\mathbf{p}$ during B_τ. This unbiased estimator is passed to the full-information regret minimization algorithm. By Lemma 6.2,

we have

$$\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - \mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right] \leq O\left(T^{2/3}|\mathcal{B}|^{1/3}(nk)^{1/3}\log^{1/3}(|\mathcal{E}|)\right)$$
$$\in O\left(T^{2/3}k^{1/3}(nk)^{1/3}(n^2k\log(nk))^{1/3}\right)$$
$$\in O\left(T^{2/3}\ nk\ \log^{1/3}(nk)\right).$$

□

7. LOWER BOUND

Can we be truly blind to the types of attackers that the defender might encounter? As mentioned before, the set of all attacker types, K, is known to the defender. But what if we allow K to include all possible types of attackers? In this section, we show that with no prior knowledge regarding the possible types of attackers that the defender might encounter, it is impossible to design a no-regret algorithm. This is formalized in our final theorem, which establishes that for any T there is a game with at most 2^{T+1} attacker types such that any online algorithm experiences regret that is linear in T.

THEOREM 7.1. *For any T there is a repeated security game in the full-information feedback setting with a set K of attacker types such that $|K| < 2^{T+1}$ and any online algorithm that at time t (possibly at random) returns strategy $\mathbf{p}_t$ has expected utility*

$$\mathbb{E}\left[\sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}_t), \mathbf{p}_t)\right] \leq \sum_{t=1}^{T} U_d(b_{a_t}(\mathbf{p}^*), \mathbf{p}^*) - \frac{T}{2},$$

where the expectation is over the algorithm's internal randomization.

PROOF. Consider a security game in which $N = \{1, 2, 3\}$, and the defender has one resource that can defend any one target at a time. Let the defender's utility be $U_d(i, \mathbf{p}) = -1$ for $i \in \{1, 3\}$ and $U_d(2, \mathbf{p}) = 0$. Let all attackers break ties in lexicographic order.

Because the defender prefers target 2 to be attacked, for any coverage probability $\mathbf{p}$, reducing p_2 to 0 leads to a valid coverage probability that will not decrease the defender's payoff. So, without loss of generality we restrict the defender's actions to the coverage probabilities given by $\mathbf{p} = [p, 0, 1-p]$.

Next, we define a set K of attackers. To do so, we first show that certain best-response functions are valid, i.e. there is an attacker — defined by its utility function — that responds according to that best-response function. The next claim, whose proof appears in Appendix A, allows us to define the attackers by their best-response functions, thereby significantly simplifying our analysis.

CLAIM 7.1.1. *For any $0 \leq r_1 \leq r_2 \leq 1$ and any $\mathbf{p} = [p, 0, 1-p]$, there exists an attacker type α_j (defined by its utility function) such that*

$$b_{\alpha_j}(\mathbf{p}) = \begin{cases} 1 & \text{if } p \in [0, r_1] \\ 2 & \text{if } p \in (r_1, r_2] \\ 3 & \text{if } p \in (r_2, 1] \end{cases}$$

Next, we recursively define $2^{T+1} - 2$ attacker types. We use $r_1 \leq r_2$ as defined in Claim 7.1.1 to represent the best response of an attacker. We represent attacker types by binary strings. Let

$$\text{Attacker } \alpha_0 : r_1^0 = \frac{1}{2}, r_2^0 = 1$$
$$\text{Attacker } \alpha_1 : r_1^1 = 0, r_2^1 = \frac{1}{2}.$$

For any $x \in \{0,1\}^{<T}$ define

$$\text{Attacker } \alpha_{0x} : r_1^{0x} = \frac{r_1^x + r_2^x}{2}, r_2^{0x} = r_2^x$$

$$\text{Attacker } \alpha_{1x} : r_1^{1x} = r_1^x, r_2^{1x} = \frac{r_1^x + r_2^x}{2}$$

This representation allows us to think of the choices of the adversary as a chain of post-fixes of a T-bit binary number. That is, the attacker chooses a T-bit binary string and plays its post-fixes in order.

Next, we formulate our problem in terms of decision trees. Consider a complete binary tree. The adversary's sequence of attackers indicates a root-to-leaf path in this tree that corresponds to his choice of the T-bit string. The defender's online decision at every step of the game is to choose a distribution over $p \in (r_1^{0x}, r_2^{0x}]$ or $p \in (r_1^{1x}, r_2^{0x}]$, having already observed string x. Since for all x, $(r_1^{0x}, r_2^{0x}] \cap (r_1^{1x}, r_2^{1x}] = \emptyset$, these two choices represent two disjoint events. Therefore, we can represent the defender's online decision at node x as a distribution on nodes $0x$ or $1x$. Using Claim 7.1.1, if both the defender and attacker land on the same node, the defender receives a penalty of 0 and otherwise receives a penalty of 1 (utility -1). To summarize, the online algorithm corresponds to a distribution over all possible decision trees; the adversary chooses a root-to-leaf path; and expected utility is calculated as above.

By Yao's Minmax Principle, an upper bound on the expected utility of the optimal randomized decision tree against the worst deterministic sequence (root-to-leaf path) can be obtained by constructing a specific distribution over sequences, and reasoning about the expected utility of the best *deterministic* decision tree against this distribution. Let this distribution be the uniform distribution over all T-bit strings. That is, at step x the adversary chooses attackers $0x$ and $1x$ each with probability $\frac{1}{2}$. Then for any fixed decision tree, at every node the algorithm only has $\frac{1}{2}$ probability of matching the adversary's choice. So, the defender receives an expected penalty of $\frac{1}{2}$. We conclude that for any randomized online algorithm there exists a sequence of attackers — corresponding to a T-bit string — such that the algorithm's expected utility is at most $-\frac{T}{2}$.

To complete the proof, we claim that for any sequence of attackers corresponding to a T-bit string, there is a mixed strategy that would cause each attacker in the sequence to attack target 2. This is true because for every x and y, $(r_1^{yx}, r_2^{yx}] \subset (r_1^x, r_2^x]$, and therefore if α_x is the attacker at step T, choosing $p^* \in (r_1^x, r_2^x]$ would also place p in the interval corresponding to any previous attacker. By Claim 7.1.1, the best response of each attacker in the sequence to the strategy $\mathbf{p}^* = [p^*, 0, 1 - p^*]$ is to attack target 2. It follows that $\mathbf{p}^*$ has overall utility 0 to the defender, and hence the regret is at least $\frac{T}{2}$. $\square$

8. DISCUSSION

Extension to general Stackelberg games. The approach presented in our work easily extends to general repeated Stackelberg games, which can be represented as follows. There are k matrices of size $n \times m$, with the same payoffs for the row player, but possibly different payoffs for the column player. This set of matrices is known to the players. At each time step, a game matrix is chosen but remains unknown to the row player. The row player then chooses a probability distribution over the rows of the matrix and the column player, having observed this distribution, chooses the column with the best expected payoff. An online algorithm should guarantee to the row player good payoff against any adversarially selected sequence of such matrices.

Note that a repeated SSG is captured by the foregoing framework: in each matrix, each row represent a pure strategy of the defender (there may be exponentially many rows) and each column represents a target. Then, for row i and column j, if target j is covered in the i^{th} strategy then the row and column payoff are $u_d^c(i)$ and $u_a^c(i)$, and otherwise $u_d^u(i)$ and $u_a^u(i)$, respectively.

To see how our methodology extends, note that each of the k matrices can be used to decompose the space of all probability distributions (over the rows) into m convex regions, where in each region

the best response is a fixed column. For the full information feedback model, the set of all extreme points in the intersections of these regions leads to an algorithm whose regret is polynomial in m, n and k, and sublinear in T. Similarly, for the partial information feedback model, our approach for inferring the frequency of each matrix by only observing the best response carries over to the general Stackelberg games. However, we focused on SSGs in order to obtain regret bounds that are polynomial in the number of targets and attacker types — the foregoing bounds for general Stackelberg games would translate to bounds that are exponential in the number of targets.

Computational complexity. The goal of our paper is to develop online algorithms that can effectively defend against an unknown sequence of attackers. We believe that the main challenge in this setting is overcoming the lack of information regarding the attackers. Therefore, it is encouraging that our algorithms achieve regret bounds that are polynomial in the number of targets and types, in addition to being sublinear in T.

Can we hope for computational tractability? A major obstacle is that many tasks involving security games are NP-Hard. For example, even computing the optimal strategy when faced with a single, known attacker type is NP-hard [Korzhyk et al. 2010]. So, in some sense, the answer to the question is negative — computational hardness is inevitable.

The good news is that the explicit representation of general Stackelberg games lends itself to polynomial-time algorithms for many tasks that are otherwise NP-hard, e.g. finding the optimal strategy in a 2-player game [Conitzer and Sandholm 2006]. Unfortunately, some of the steps in our algorithms remain computationally inefficient even for general repeated Stackelberg games. In particular, keeping track of the loss of all extreme points and computing a barycentric spanning set over exponentially many regions both require exponential time. It would be interesting to find out whether there exist computationally efficient no-regret algorithms for general repeated Stackelberg games.

On the power of adversary. In our work, we assume that the sequence of attackers is chosen adversarially before the game starts. That is, the adversary's choice of attacker is *oblivious* to the history. It is worth noting that our upper bound on regret in the full-information feedback setting holds for a much more powerful adversary; an *adaptive* adversary who chooses an attacker at time t by first observing the defender's mixed strategies $\mathbf{p}_1, \ldots, \mathbf{p}_{t-1}$, and the defender's distribution (determined by the internal randomness of the online algorithm) over mixed strategies at steps $1, \ldots, t$. It would be interesting to know whether there are no-regret algorithms, with polynomial dependence on the number of targets and types, for adaptive adversaries and partial-information feedback.

On the benefits of laziness. Our regret analysis holds when used in conjunction with any regret-minimization algorithm that satisfies Proposition 1. Nevertheless, some algorithms provide additional properties that may prove useful in practice. Specifically, when used with *Follow the Lazy Leader*, our algorithm for the full information setting uses one mixed strategy for a long period of time before switching to another (expected length $\tilde{O}(\sqrt{T})$). Informally speaking, this allows attackers enough time to conduct surveillance, observe the mixed strategy, and then best-respond to it. Therefore, even if the attacker's response to a new mixed strategy is unreliable or irrational at first (not representative of his best response), the defender can still guarantee a no-regret outcome.

The extremely partial information model. Previous work has mostly assumed that the defender is able to monitor all targets simultaneously and detect an attack on any one of them at all times [Kiekintveld et al. 2011; Marecki et al. 2012]. In contrast, our algorithm for the partial information setting only requires the ability to detect whether or not an attack occurs on at most *one chosen target* at a time. This feature may prove useful in domains where simultaneous observation of targets is impractical. For example, in wildlife protection applications, patrols and unmanned aerial vehicles (UAVs) can detect signs of poaching only in very specific areas.

Other sources of uncertainty. In our work, as is common in other theoretical papers on SSGs, we considered attackers with perfect observation and rationality. That is, we assume the attacker can

(eventually, as in the case of Follow the Lazy Leader) observe the mixed strategy of the defender *perfectly* and then choose the target with *absolute highest* expected utility. Existing work in one-shot security games studies attackers with limited surveillance [An et al. 2012; Blum et al. 2014a] and bounded rationality [Jiang et al. 2013; Yang et al. 2014], and in some cases provides theoretical guarantees on the quality of solutions. It would be interesting to know whether there are no-regret algorithms that handle all of these different uncertainties, simultaneously.

REFERENCES

AN, B., KEMPE, D., KIEKINTVELD, C., SHIEH, E., SINGH, S. P., TAMBE, M., AND VOROBEYCHIK, Y. 2012. Security games with limited surveillance. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*. 1242–1248.

AUER, P., CESA-BIANCHI, N., FREUND, Y., AND SCHAPIRE, R. E. 1995. Gambling in a rigged casino: The adversarial multi-armed bandit problem. In *Proceedings of the 36th Symposium on Foundations of Computer Science (FOCS)*. 322–331.

AWERBUCH, B. AND KLEINBERG, R. 2008. Online linear optimization and adaptive routing. *Journal of Computer and System Sciences 74,* 1, 97–114.

AWERBUCH, B. AND MANSOUR, Y. 2003. Adapting to a reliable network path. In *Proceedings of the 22nd Annual Symposium on Principles of Distributed Computing (PODC)*. 360–367.

BLUM, A., HAGHTALAB, N., AND PROCACCIA, A. D. 2014a. Lazy defenders are almost optimal against diligent attackers. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence (AAAI)*. 573–579.

BLUM, A., HAGHTALAB, N., AND PROCACCIA, A. D. 2014b. Learning optimal commitment to overcome insecurity. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS)*. 1826–1834.

BLUM, A. AND MANSOUR, Y. 2007. Learning, regret minimization, and equilibria. In *Algorithmic Game Theory*, N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, Eds. Cambridge University Press, Chapter 4.

BUBECK, S. AND CESA-BIANCHI, N. 2012. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *CoRR abs/1204.5721*.

CESA-BIANCHI, N., MANSOUR, Y., AND STOLTZ, G. 2007. Improved second-order bounds for prediction with expert advice. *Machine Learning 66,* 2–3, 321–352.

CONITZER, V. AND SANDHOLM, T. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM Conference on Economics and Computation (EC)*. 82–90.

JIANG, A. X., NGUYEN, T. H., TAMBE, M., AND PROCACCIA, A. D. 2013. Monotonic maximin: A robust Stackelberg solution against boundedly rational followers. In *Proceedings of the 4th Conference on Decision and Game Theory for Security (GameSec)*. 119–139.

KALAI, A. AND VEMPALA, S. 2005. Efficient algorithms for online decision problems. *Journal of Computer and System Sciences 71,* 3, 291–307.

KIEKINTVELD, C., MARECKI, J., AND TAMBE, M. 2011. Approximation methods for infinite Bayesian Stackelberg games: Modeling distributional payoff uncertainty. In *Proceedings of the 10th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 1005–1012.

KORZHYK, D., CONITZER, V., AND PARR, R. 2010. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*. 805–810.

LETCHFORD, J., CONITZER, V., AND MUNAGALA, K. 2009. Learning and approximating the optimal strategy to commit to. In *Proceedings of the 2nd International Symposium on Algorithmic Game Theory (SAGT)*. 250–262.

LITTLESTONE, N. AND WARMUTH, M. K. 1994. The weighted majority algorithm. *Information and computation 108,* 2, 212–261.

MARECKI, J., TESAURO, G., AND SEGAL, R. 2012. Playing repeated Stackelberg games with unknown opponents. In *Proceedings of the 11th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 821–828.

PITA, J., JAIN, M., TAMBE, M., ORDÓÑEZ, F., AND KRAUS, S. 2010. Robust solutions to Stackelberg games: Addressing bounded rationality and limited observations in human cognition. *Artificial Intelligence 174,* 15, 1142–1171.

TAMBE, M. 2012. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

YANG, R., FORD, B. J., TAMBE, M., AND LEMIEUX, A. 2014. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the 13th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. 453–460.

ZINKEVICH, M. 2003. Online convex programming and generalized infinitesimal gradient ascent. In *Proceedings of the 20th International Conference on Machine Learning (ICML)*. 928–936.

Online Appendix to: Commitment Without Regrets: Online Learning in Stackelberg Security Games

Maria-Florina Balcan, Carnegie Mellon University
Avrim Blum, Carnegie Mellon University
Nika Haghtalab, Carnegie Mellon University
Ariel D. Procaccia, Carnegie Mellon University

A. PROOF OF CLAIM 7.1.1

Let $\|u\| = \max\{\frac{1-r_1}{r_1}, \frac{r_2}{1-r_2}\}$. Let α_j be an attacker defined as follow:

$$u^c_{\alpha_j}(i) = \begin{cases} \frac{-(1-r_1)}{r_1 \|u\|} & \text{if } i = 1 \\ 0 & \text{if } i = 2 \\ \frac{-r_2}{(1-r_2)\|u\|} & \text{if } i = 3 \end{cases} \quad \text{and} \quad u^u_{\alpha_j}(i) = \begin{cases} 0 & \text{if } i = 2 \\ \frac{1}{\|u\|} & \text{otherwise} \end{cases}$$

We show that attacker α_j's best-response has the desired properties. For $p \in [0, r_1]$,

$$\begin{aligned} U_{\alpha_j}(1, \mathbf{p}) &= \frac{1}{\|u\|}\left(p\frac{-(1-r_1)}{r_1} + (1-p)\right) \geq \frac{1}{\|u\|}(-(1-r_1) + (1-p)) \geq 0, \\ U_{\alpha_j}(2, \mathbf{p}) &= 0, \\ U_{\alpha_j}(3, \mathbf{p}) &= \frac{1}{\|u\|}\left((1-p)\frac{-r_2}{1-r_2} + p\right) \leq \frac{1}{\|u\|}(-r_2 + p) \leq 0, \end{aligned}$$

so, $b_{\alpha_j}(\mathbf{p}) = 1$ for $p \in [0, r_1]$. For $p \in (r_1, r_2]$,

$$\begin{aligned} U_{\alpha_j}(2, \mathbf{p}) = 0 &> \frac{1}{\|u\|}\left(-(1-r_1) + (1-p)\right) > \frac{1}{\|u\|}\left(p\frac{-(1-r_1)}{r_1} + (1-p)\right) = U_{\alpha_j}(1, \mathbf{p}), \\ U_{\alpha_j}(2, \mathbf{p}) = 0 &\geq \frac{1}{\|u\|}\left(-r_2 + p\right) \geq \frac{1}{\|u\|}\left((1-p)\frac{-r_2}{1-r_2} + p\right) = U_{\alpha_j}(3, \mathbf{p}), \end{aligned}$$

so, $b_{\alpha_j}(\mathbf{p}) = 2$ for $p \in (r_1, r_2]$. For $p \in (r_2, 1]$,

$$\begin{aligned} U_{\alpha_j}(1, \mathbf{p}) &= \frac{1}{\|u\|}\left(p\frac{-(1-r_1)}{r_1} + (1-p)\right) < \frac{1}{\|u\|}\left(-(1-r_1) + (1-p)\right) < 0, \\ U_{\alpha_j}(2, \mathbf{p}) &= 0, \\ U_{\alpha_j}(3, \mathbf{p}) &= \frac{1}{\|u\|}\left((1-p)\frac{-r_2}{1-r_2} + p\right) > \frac{1}{\|u\|}\left(-r_2 + p\right) > 0, \end{aligned}$$

so, $b_{\alpha_j}(\mathbf{p}) = 3$. Therefore, the attacker defined by the above utility function has the desired best-response.

DOI: http://dx.doi.org/10.1145/0000000.0000000