Signaling in Bayesian Stackelberg Games

Haifeng Xu[1], **Rupert Freeman**[2], **Vincent Conitzer**[2], **Shaddin Dughmi**[1], **Milind Tambe**[1]

[1]University of Southern California, Los Angeles, CA 90007, USA
{haifengx,shaddin,tambe}@usc.edu

[2]Duke University, Durham, NC 27708, USA
{rupert,conitzer}@cs.duke.edu

ABSTRACT

Algorithms for solving Stackelberg games are used in an ever-growing variety of real-world domains. Previous work has extended this framework to allow the leader to commit not only to a distribution over actions, but also to a scheme for stochastically signaling information about these actions to the follower. This can result in higher utility for the leader. In this paper, we extend this methodology to Bayesian games, in which either the leader or the follower has payoff-relevant private information or both. This leads to novel variants of the model, for example by imposing an incentive compatibility constraint for each type to listen to the signal intended for it. We show that, in contrast to previous hardness results for the case without signaling [5, 16], we can solve unrestricted games in time polynomial in their natural representation. For security games, we obtain hardness results as well as efficient algorithms, depending on the settings. We show the benefits of our approach in experimental evaluations of our algorithms.

Keywords

Bayesian Stackelberg Games, Algorithms, Signaling, Security Games

1. INTRODUCTION

In the algorithmic game theory community, and especially the multiagent systems part of that community, there has been rapidly increasing interest in Stackelberg models where the leader can commit to a mixed strategy. This interest is driven in part by a number of high-impact deployed security applications [25]. One of the advantages of this framework—as opposed to, say, computing a Nash equilibrium of the simultaneous-move game—is that it sidesteps issues of equilibrium selection. Another is that in two-player normal-form games, an optimal mixed strategy to commit to can be found in polynomial time [5]. There are limits to this computational advantage, however; once we extend to three-player games or Bayesian games, the computational problem becomes hard again [5]. (In a Bayesian game, some of the players have private information that is relevant to the payoffs; their private information is encoded by their *type*.)

As has previously been observed [4, 28, 23], the leader may be able to do more than commit to a mixed strategy. The leader may additionally be able to commit to send signals to the follower(s) that are correlated with the action she has chosen. This ability can of course never hurt the leader: she always has the choice of sending an uninformative signal. In a two-player normal-form game, it turns out that no benefit can be had from sending an informative signal. This is because the expected leader utility conditioned on each signal, which corresponds to a posterior belief of the leader's action, is weakly dominated by the expected leader utility of committing to the optimal mixed strategy [4]. But this is no longer true in games with three or more players. Moreover, intriguingly, the enriched problem with signaling can still be solved in polynomial time in these games [4]. The idea of adding signals has also already been explored in security games [28], however these games were not Bayesian (but with richer game structure).

In this paper, we extend this line of work to Bayesian Stackelberg Games (**BSG**s). We suppose that, when the follower has multiple possible types, the leader is able to send a separate signal to each of these types, without learning what the type is. For example, consider a security game on a rail network in which we aim to catch ticketless travelers (or, better yet, give them incentives to buy a ticket). Here, the attacker's type could encode at which location he starts his journey. Then, by making a separate announcement at each station, we send a separate signal to each type. As another example, we may send different signals over different (say) radio frequencies. In this case, each follower type receives a separate signal depending on the frequency to which he is listening. In this latter example (unlike the former), we also require an *incentive compatibility (IC)* constraint: no type should find it beneficial to switch over to a different frequency, since we have no way of forcing a type to listen to a particular frequency.

Besides considering the case of multiple follower types, we also consider the case of multiple leader types. Here, the signal sent by the leader can be correlated with her type as well as her action. Among other examples, this allows us to capture models where the leader is a seller of some item, and the type of the leader corresponds to knowledge about, for example, the quality of the item. She can then send an informative (but perhaps not *completely* informative) signal about this quality to the buyer. Such models are sometimes studied in the auction design literature [8, 18, 12], but here our interest is in generally applicable algorithms.

Our Contributions: We consider signaling in different models of Bayesian Stackelberg games, and essentially pin down the computational complexity in each. For the case with multiple follower types (but a single leader type), we show that the optimal combinations of mixed strategies and signaling schemes can be computed in polynomial time using linear programming.[1] This is the case

[1]One may wonder whether this just follows from the fact that we can model Bayesian games by representing each type as a single player, thereby reducing it to a multiplayer game. But this does not

Appears in: *Proceedings of the 15th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2016), J. Thangarajah, K. Tuyls, C. Jonker, S. Marsella (eds.), May 9–13, 2016, Singapore.*

whether an incentive compatibility constraint applies or not. However, for security games, we show that the problem is NP-hard, though we do identify a special case that can be solved efficiently. We also provide hardness evidence that this special case is almost the best one can hope for in terms of polynomial computability.

For the case with multiple leader types (but a single follower type), we show that the optimal combinations of mixed strategies and signaling schemes can also be computed in polynomial time. Moreover, the polynomial-time solvability extends to security games in this setting. We note that our results (both hardness and polynomial-time solvability) can be easily generalized to the case with *both* multiple leader and follower types, thus we will not discuss it explicitly in the paper. We conclude with an experimental evaluation of our approach.

2. AN EXAMPLE OF STACKELBERG COMPETITION

The Stackelberg model was originally introduced to capture market competition between a *leader* (e.g., a leading firm in some area) and a *follower* (e.g., an emerging start-up). The leader has an advantage of committing to a strategy (or equivalently, moving first) before the follower makes decisions. Here we consider a Bayesian case of Stackelberg competition where the leader does not have full information about the follower.

For example, consider a market with two firms, a leader and a follower. The leader specializes in two products, product 1 and product 2. The follower is a new start-up which focuses on only *one* product. It is publicly known that the follower will focus on product 1 with probability 0.55 (call him a follower of type θ_1 in this case), and product 2 with probability 0.45 (call him a follower of type θ_2). But the realization is only known to the follower. The leader has a research team, and must decide which product to devote this (indivisible) team to, or to send them on vacation. On the other hand, the follower has two options: either entering the market and developing the product he focuses on, or leaving the market.

Naturally, the follower wants to avoid competition with the leader's research team. In particular, depending on the type of the follower, the leader's decision may drive the follower out of the market or leave the follower with a chance to gain substantial market share. This can be modeled as a Bayesian Stackelberg Game (BSG) where the leader has one type and the follower has two possible types. To be concrete, we specify the payoff matrices for different types of follower in Figure 1, where the leader's action L_i simply denotes the leader's decision to devote the team to product i for $i \in \{1, 2, \emptyset\}$; $\emptyset$ means a team vacation. Similarly, the follower's action F_i means the follower focuses on products $i \in \{1, 2, \emptyset\}$ where $\emptyset$ means leaving the market. Notice that the payoff matrices force the follower to only produce the product that is consistent with his type, otherwise he gets utility $-\infty$. The utility for the leader is relatively simple: the leader gets utility 1 only if the follower (of any type) takes action $F_\emptyset$, i.e., leaving the market, and gets utility 0 otherwise. In other words, the leader wants to drive the follower out of the market.

Possessing a first-mover advantage, the leader can commit to a *randomized* strategy to assign her research team so that it maximizes her utility in expectation over the randomness of her mixed strategy and the follower types. Unfortunately, finding the optimal mixed strategy to commit to turns out to be NP-hard for BSGs in general [5]. Nevertheless, by exploiting the special structure in this example, it is easy to show that any mixed strategy that puts at least

work, because the corresponding normal form of the game would have size exponential in the number of types.

	$F_\emptyset$	F_1	F_2
$L_\emptyset$	0	2	$-\infty$
L_1	0	-1	$-\infty$
L_2	0	2	$-\infty$

type θ_1, $p = 0.55$

	$F_\emptyset$	F_1	F_2
$L_\emptyset$	0	$-\infty$	1
L_1	0	$-\infty$	1
L_2	0	$-\infty$	-1

type θ_2, $p = 0.45$

Figure 1: Payoff Matrices for Followers of Different Types

$2/3$ probability on L_1 is optimal for the leader to commit to. This is because to drive a follower of type θ_1 out of the market, the leader has to take L_1 with probability at least $2/3$. Likewise, to drive a follower of type θ_2 out of the market, the leader has to take L_2 with probability at least $1/2$. Since $2/3 + 1/2 > 1$, the leader cannot achieve both, so the optimal choice is to drive the follower of type θ_1 (occurring with a higher probability) out of the market so that the leader gets utility 0.55 in expectation.

Notice that the leader commits to the strategy without knowing the realization of the follower's type. This is reasonable because the follower, as a start-up, can keep information confidential from the leader firm at the initial stage of the competition. However, as time goes on, the leader will gradually learn the type of the follower. Nevertheless, the leader firm cannot change her chosen action at that point because, for example, there is insufficient time to switch to another product. Can the leader still do something strategic at this point? In particular, we study whether the leader can benefit by partially revealing her action to the follower after observing the follower's type. To be concrete, consider the following leader policy. Before observing the follower's type, the leader commits to choose action L_1 and L_2 uniformly at random, each with probability $1/2$. Meanwhile, the leader also commits to the following *signaling scheme*. If the follower has type θ_1, the leader will send a signal $\sigma_\emptyset$ to the follower when the leader takes action L_1, and will send either $\sigma_\emptyset$ or σ_1 uniformly at random when the leader takes action L_2. Mathematically, the signaling scheme for the follower of type θ_1 is captured by the following probabilities.

$$\mathbf{Pr}(\sigma_\emptyset | L_1, \theta_1) = 1 \quad \mathbf{Pr}(\sigma_1 | L_1, \theta_1) = 0;$$
$$\mathbf{Pr}(\sigma_\emptyset | L_2, \theta_1) = \tfrac{1}{2} \quad \mathbf{Pr}(\sigma_1 | L_2, \theta_1) = \tfrac{1}{2}.$$

On the other hand, if the follower has type θ_2, the leader will always send $\sigma_\emptyset$ regardless of what action she has taken.

When a follower of type θ_1 receives signal $\sigma_\emptyset$ (occurring with probability $3/4$), he infers the posterior belief of the leader's strategy as $\mathbf{Pr}(L_1 | \sigma_\emptyset, \theta_1) = 2/3$ and $\mathbf{Pr}(L_2 | \sigma_\emptyset, \theta_1) = 1/3$, thus deriving an expected utility of 0 from taking action F_1. Assuming the follower breaks ties in favor of the leader,[2] he will then choose action $F_\emptyset$, leaving the market. On the other hand, if the follower receives σ_1 (occurring with probability $1/4$), he knows that the leader has taken action L_2 for sure; thus the follower will take action F_1, achieving utility 2. In other words, the signals $\sigma_\emptyset$ and σ_1 can be viewed as recommendations to the follower to leave the market ($\sigma_\emptyset$) or develop the product (σ_1), though we emphasize that a signal has no meaning beyond the posterior distribution on leader's actions that it induces. As a result, the leader drives the follower out of the market $3/4$ of the time. On the other hand, if the follower has type θ_2, since the leader reveals no information, the follower derives expected utility 0 from taking F_2, and thus will choose F_0 in favor of the leader. In expectation, the leader gets utility $\frac{3}{4} \times \frac{1}{2} + \frac{1}{2} = 0.875 (> 0.55)$. Thus, the leader achieves better utility by signaling.

The design of the signaling scheme above depends crucially on the fact that the leader can distinguish different follower types be-

[2]This is without loss of generality because the leader can always slightly tune the probability mass to make the follower slightly prefer $F_\emptyset$.

fore sending the signals and will signal differently to different follower types. This fits the setting where the leader can observe the follower's type after the leader takes her action and then signals accordingly. However, in many cases, the leader is *not* able to observe the follower's type. Interestingly, it turns out that the leader can in some cases design a signaling scheme which incentivizes the follower to *truthfully* report his type to the leader and still benefit from signaling. Note that the signaling scheme above does not satisfy the follower's incentive compatibility constraints – if the follower is asked to report his type, a follower of type θ_2 would be better off to report his type as θ_1. This follows from some simple calculation, but an intuitive reason is that a follower of type θ_2 will not get any information if he truthfully reports θ_2, but will receive a more informative signal, thus benefit himself, by reporting θ_1.

Now let us consider another leader policy. The leader commits to the mixed strategy $(L_\emptyset, L_1, L_2) = (1/11, 6/11, 4/11)$. Interestingly, this involves sometimes sending the research team on vacation! Meanwhile, the leader also commits to the following more sophisticated signaling scheme. If the follower reports type θ_1, the leader will send signal $\sigma_\emptyset$ whenever L_1 is taken as well as $\frac{3}{4}$ of the time that L_2 is taken; otherwise the leader sends signal σ_1. If the follower reports type θ_2, the leader sends signal $\sigma_\emptyset$ whenever L_2 is taken as well as $\frac{2}{3}$ of the time that L_1 is taken; otherwise the leader sends signal σ_2. It turns out that this policy is incentive compatible – truthfully reporting the type is in the follower's best interests – and achieves the maximum expected leader utility $\frac{17}{22} \approx 0.773 \in (0.55, 0.875)$ among all such policies.

Justification of Commitment: The assumption of commitment to strategies is well motivated, and has been justified, in many applications, e.g., market competition [9] and security [25]. This is usually due to the leader's first-mover advantage. The assumption of commitment to signaling schemes is justified on the grounds of games that are played repeatedly (e.g., a leading firm plays repeatedly with start-ups that can show up and fade away), so the follower can learn the signaling scheme - how the signals correlate with leader actions taken. On the other hand, to balance the short term utility and long-term credibility, the leader has incentives to follow the signaling scheme in order to build a reputation about her strategy of disclosing information. We refer the reader to [24] for more thorough discussions of this phenomenon.

Remark: This example shows that the additional ability of committing to a signaling scheme can profoundly affect both players' strategies. We study how such additional commitment changes the game as well as the computation of the leader's optimal policy. The rest of this paper is organized as follows. In Section 3 we generalize the above example to BSGs, and also examine its application to Bayesian Stackelberg *Security Games*, a model of growing interest in modeling various security challenges. Note that the above example only concerned the case where the *follower* has multiple types. In Section 4, we consider a variant of the model where the *leader* has multiple types (but the follower has only one type), and seek to compute the optimal leader policy. We show simulation results in Section 5 and conclude in Section 6.

3. SINGLE LEADER TYPE, MULTIPLE FOLLOWER TYPES

3.1 The Model

In this section, we generalize the example in Section 2 and consider how the leader's additional ability of committing to a signaling scheme changes the game and the computation. We start with a Bayesian Stackelberg Game (**BSG**) with one *leader* type and multiple *follower* types. Let Θ denote the set of all the follower types. An instance of such a BSG is given by a set of tuples $\{(A^\theta, B^\theta, \lambda_\theta)\}_{\theta\in\Theta}$ where $A^\theta, B^\theta \in \mathbb{R}^{m\times n}$ are the payoff matrices of the leader (row player) and the follower (column player) respectively when the follower has type θ, which occurs with probability λ_θ. We use $[m]$ and $[n]$ to denote the leader's and follower's pure strategy set respectively. For convenience, we assume that every follower type has the same number of actions (i.e., n) in the above notation. This is without loss of generality since we can always add "dummy" actions with payoff $-\infty$ to both players. We use a^θ_{ij} $[b^\theta_{ij}]$ to denote a generic entry of A^θ $[B^\theta]$. If $A^\theta = -B^\theta$ for all $\theta \in \Theta$, we say that the BSG is *zero-sum*. Following the standard assumption of Stackelberg games, we assume that the leader can commit to a mixed strategy. Such a leader strategy is *optimal* if it results in maximal leader utility in expectation over the randomness of the strategy and follower types, assuming each follower type best responds to the leader's mixed strategy.[3] It is known that computing the optimal mixed strategy, also known as the Bayesian Strong Stackelberg Equilibrium (**BSSE**) strategy, to commit to is NP-hard in such a BSG [5]. A later result strengthened the hardness to approximation – no polynomial time algorithm can give a non-trivial approximation ratio in general unless P=NP [16].

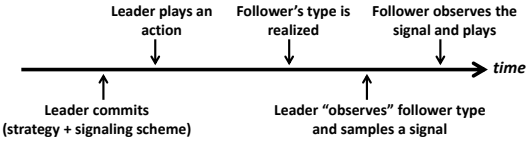


Figure 2: Timeline of the BSG with Multiple Follower Types.

We consider a richer model where the defender can commit not only to a mixed strategy but also to a scheme, often known as a *signaling scheme*, of partially releasing information regarding the action she is currently playing i.e., the sample from the leader's committed mixed strategy. Formally, the leader commits to a mixed strategy $\mathbf{x} \in \Delta_m$, where Δ_m is the m-dimensional simplex, and a signaling scheme φ which is a *stochastic* map from $\Theta \times [m]$ to a set of signals Σ. In other words, the sender randomly chooses a signal to send based on the action she currently plays and the follower type she observes. We call the pair

$$(\mathbf{x}, \varphi) \text{ where } \mathbf{x} \in \Delta_m;\ \varphi : \Theta \times [m] \xrightarrow{rnd} \Sigma \tag{1}$$

a *leader policy*. After the commitment, the leader samples an action to play. Then the follower's type is realized, and the leader observes the follower's type and samples a signal. We assume that the follower has full knowledge of the leader policy. Upon receiving a signal, the follower updates his belief about the leader's action and takes a best response. Figure 2 illustrates the timeline of the game.

We note that if the leader cannot distinguish different follower types and has to send the same signal to all different follower types, then signaling does not benefit the leader (for the same reason as the non-Bayesian setting). In this case, she should simply commit to the optimal mixed strategy. The leader only benefits when she can target different follower types with different signals. In many cases, like the example in Section 2, the leader gets to observe the follower's type when it is realized (but after her action is completed) and can therefore choose to signal differently to different follower types. Moreover, in practice it is sometimes natural for the leader to send different signals to different follower types even without gen-

[3]Note that the follower cannot observe the leader's realized action, which is a standard assumption in Stackelberg games.

uinely learning their types, e.g., the follower's type may be defined by their location, in which case we can send signals using location-specific devices such as physical signs or radio transmission – this fits our model just as well. We will elaborate one such example when discussing security games.

3.2 Commitment to Optimal Leader Policy

We first consider the case where the leader can explicitly observe the follower's type, and thus can signal differently to different follower types, but this would also fit the location based model. We start with a simple observation.

OBSERVATION 3.1 (SEE, E.G., [15]). *There exists an optimal signaling scheme using at most n signals with signal σ_j recommending action $j \in [n]$ to the follower.*

Observation 3.1 follows simply from the fact that if two signals result in the same follower best-response action, we can merge these signals, resulting in a new signal without changing the follower's best response action and the leader's utility. As a result, for the rest of the paper we assume that $\Sigma = \{\sigma_j\}_{j\in[n]}$.

THEOREM 3.2. *The optimal leader policy can be computed in $poly(m, n, |\Theta|)$ time by linear programming.*

PROOF. Let $\mathbf{x} = (x_1, ..., x_m) \in \Delta_m$ be the leader's mixed strategy to commit to. As a result of Observation 3.1, the signaling scheme φ can be characterized by $\varphi(j|i, \theta)$ which is the probability of sending signal σ_j conditioned on the leader's (pure) action i and the follower's type θ. Then, $p^\theta_{ij} = x_i \cdot \varphi(j|i, \theta)$ is the *joint probability* that the leader plays pure strategy i and sends signal σ_j, conditioned on observing the follower of type θ. Then the following linear program computes the optimal leader policy captured by variables $\{x_i\}_{i\in[m]}$ and $\{p^\theta_{ij}\}_{i\in[m],j\in[n],\theta\in\Theta}$.

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta} \lambda_\theta \sum_{ij} p^\theta_{ij} a^\theta_{ij} & \\ \text{subject to} & \sum_{j=1}^n p^\theta_{ij} = x_i, & \text{for } i \in [m], \theta \in \Theta. \\ & \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij} \geq \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij'}, & \text{for } \theta, j \neq j'. \\ & \sum_{i=1}^m x_i = 1 & \\ & p^\theta_{ij} \geq 0, & \text{for all } i, j, \theta. \end{array} \tag{2}$$

.

The first set of constraints mean that the summation of probability mass p^θ_{ij} – the joint probability of playing pure strategy i and sending signal σ_j conditioned on follower type θ – over j should equal the probability of playing action i for any type θ. The second set of constraints are to guarantee that the recommended action j by signal σ_j is indeed the follower's best response.[4] □

Given any game G, let $U_{sig}(G)$ be the leader's expected utility by taking the optimal leader policy computed by LP (2). Moreover, let $U_{BSSE}(G)$ be the leader's utility in the BSSE, i.e., the expected leader utility by committing to (only) the optimal mixed strategy.

PROPOSITION 3.3. *If G is a zero-sum BSG, then $U_{sig}(G) = U_{BSSE}(G)$. That is, the leader does not benefit from signaling in zero-sum BSGs.*

The intuition underlying Proposition 3.3 is that, in a situation of pure competition, any information volunteered to the follower will be used to "harm" the leader. In other words, signaling is only helpful when the game exhibits some "cooperative components". We defer the formal proof to the appendix at the end of this paper.

Remark: Notice that computing the optimal mixed strategy (assuming no signaling) to commit to is NP-hard in general for the setting above (even NP-hard to approximate within any non-trivial ratio), as shown in [5, 16]. Interestingly, it turns out that when we consider a richer model with signaling, the problem becomes easy! Intuitively, this is because the signaling scheme "relaxes" the game by introducing correlation between the leader's and follower's action (via the signal). Such correlation allows more efficient computation. Similar intuition can be seen in the literature on computing Nash equilibria (hard for two players [6, 3]) and correlated equilibria (easy in fairly general settings [20, 14]).

[4]This is often called "obedience".

3.3 Incentivizing the Follower Type

In many situations, it is not realistic to expect that the leader can observe the follower's type. For example, the follower's type may be whether he has a high or low value for an object, which is not directly observable. In such cases, the leader can ask the follower to report his type. However, it is not always in the follower's best interests to *truthfully* report his own type since the signal that is intended for a different follower type might be more beneficial to the follower (recall the example in Section 2). In this section, we consider how to compute an optimal *incentive compatible* (**IC**) leader policy that incentivizes the follower to truthfully report his type, and meanwhile benefits the leader.

We note that Observation 3.1 still holds in this setting. To see this, consider a follower of type θ that receives more than one signal, each resulting in the same follower best response. Then, as before, we can merge these signals without harming the follower of type θ. But if a follower of type $\beta \neq \theta$ misreports his type as θ, receiving the merged signal provides less information than receiving one of the unmerged signals. Therefore, if the follower of type β had no incentive to misreport type θ before the signals were merged, he has no incentive to misreport after the signals are merged. So any signaling scheme with more than n signals can be reduced to an equivalent scheme with exactly n signals.

THEOREM 3.4. *The optimal* incentive compatible *(***IC***) leader policy can be computed in $poly(m, n, |\Theta|)$ time by linear programming, assuming the leader does* not *observe the follower's type.*

PROOF. Similar to Section 3.2, we still use variables $\mathbf{x} \in \Delta_m$ and $\{p^\theta_{ij}\}_{i\in[m],j\in[n],\theta\in\Theta}$ to capture the leader's policy. Then $\alpha^\theta_j = \sum_{i=1}^m p^\theta_{ij}$ is the probability of sending signal j when the follower has type θ. Now consider the case where the follower reports type β, but has true type θ. When the leader recommends action j (assuming a follower of type β), which now is *not* necessarily the follower's best response due to the follower's misreport, the follower's utility for any action j' is $\frac{1}{\alpha^\beta_j} \sum_{i=1}^m p^\beta_{ij} b^\theta_{ij'}$. Therefore, the follower's action will be $\arg\max_{j'} \frac{1}{\alpha^\beta_j} \sum_{i=1}^m p^\beta_{ij} b^\theta_{ij'}$ with expected utility $\max_{j'} \frac{1}{\alpha^\beta_j} \sum_{i=1}^m p^\beta_{ij} b^\theta_{ij'}$. As a result, the expected utility for the follower of type θ, but misreporting type β, is

$$U(\beta; \theta) = \sum_{j=1}^n \left[\alpha^\beta_j \times \max_{j'} \frac{1}{\alpha^\beta_j} \sum_{i=1}^m p^\beta_{ij} b^\theta_{ij'} \right] = \sum_{j=1}^n \left[\max_{j'} \sum_{i=1}^m p^\beta_{ij} b^\theta_{ij'} \right]$$

Therefore, to incentivize the follower to truthfully report his type, we only need to add the incentive compatibility constraints $U(\theta; \theta) \geq U(\beta; \theta)$. Using the condition $\max_{j'} \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij'} = \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij}$, i.e., the recommended action j by σ_j is indeed the follower's best response when the follower has type θ, we have

$$U(\theta; \theta) = \sum_{j=1}^n \left[\max_{j'} \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij'} \right] = \sum_{j=1}^n \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij}$$

Therefore, incorporating the above constraints to LP (2) gives the following optimization program which computes an optimal incen-

tive compatible leader policy.

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta}\lambda_\theta \sum_{ij} p^\theta_{ij} a^\theta_{ij} & \\ \text{subject to} & \sum_{j=1}^n p^\theta_{ij} = x_i, & \text{for all } i,\theta. \\ & \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij} \geq \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij'}, & \text{for } j \neq j'. \\ & \sum_{j=1}^n \sum_{i=1}^m p^\theta_{ij} b^\theta_{ij} \geq & \\ & \quad \sum_{j=1}^n \left[\max_{j'} \sum_{i=1}^m p^\beta_{ij} b^\theta_{ij'}\right], & \text{for } \beta \neq \theta. \\ & \sum_{i=1}^m x_i = 1 & \\ & p^\theta_{ij} \geq 0, & \text{for all } i,j,\theta. \end{array} \tag{3}$$

Notice that $\sum_{j=1}^n \left[\max_{j'} \sum_{i=1}^m p^\beta_{ij} b^\theta_{ij'}\right]$ is a convex function, therefore the above is a convex program. By standard tricks, the convex constraint can be converted to a set of polynomially many linear constraints (see, e.g., [2]). □

Given any BSG G, let $U_{IC}(G)$ be the expected leader utility by playing an optimal incentive compatible leader policy computed by Convex Program (3). The following theorem captures the utility ranking of the different models.

PROPOSITION 3.5 (UTILITY RANKING).

$$U_{sig}(G) \geq U_{IC}(G) \geq U_{BSSE}(G).$$

PROOF. The first inequality holds because any feasible solution to Program (3) must also be feasible to LP (2). The second inequality follows from the fact that the BSSE is an incentive compatible leader policy where the signaling scheme simply reveals no information to any follower. This scheme is trivially incentive compatible because it is indifferent to the follower's report. □

Relation to Other Models. Our model in this section relates to the model of Persuasion with Privately Informed Receivers ("followers" in our terminology) by Kolotilin et al. [1]. Though in a different context, the model of Kolotilin et al. is essentially a BSG played between a leader and a follower of type only known to himself. In our model, players' payoffs are affected by the leader's action, thus the leader first commits to a mixed strategy and then signals her sampled action to the follower with incentive compatibility constraints. In [1], the leader does not have actions. Instead, the payoffs are determined by some random state of nature, which the leader can privately observe but does not have control over. The follower only has a prior belief about the state of nature, analogous to the follower knowing the leader's mixed strategy in our model. Kolotilin et al. study how the leader can signal such exogenously given information to the follower with incentive compatibility constraints. Mathematically, this corresponds to the case where $\mathbf{x}$ in Program (3) is given *a-priori* instead of being designed.

3.4 Security Games

In this section we consider the Bayesian *Security Games* played between a defender (leader) and an attacker (follower). Our results here are generally *negative* – the optimal leader policy becomes hard to compute even in the simplest of the security games. In particular, we consider a security game with n targets and k $(< n)$ *identical unconstrained* security resources. Each resource can be assigned to at most one target; a target with a resource assigned is called *covered*, otherwise it is *uncovered*. Therefore, the defender pure strategies are subsets of targets (to be protected) of cardinality k. On the other hand, the attacker has n actions – attack any one of the n targets. The attacker has a private type θ which is drawn from finite set Θ with probability λ_θ. The attacker is privy to his own type, but the defender only knows the distribution $\{\lambda_\theta\}_{\theta\in\Theta}$. This captures many natural security settings. For example, in airport patrolling, the attacker could either be a terrorist or a regular policy violator as modeled in [22]. In wildlife patrolling, the type of an attacker could be the species the attacker is interested in [10]. If the attacker chooses to attack target $i \in [n]$, players' utilities depend not only on whether target i is covered or not, but also on the attacker's type θ. We use $U^{d/a}_{c/u}(i|\theta)$ to denote the defender/attacker (d/a) utility when target i is covered/uncovered (c/u) and an attacker of type θ attacks target i.

Notice that the leader now has $\binom{n}{k}$ pure strategies, thus the natural LP has exponential size. Nevertheless, in security games we can sometimes solve the game efficiently by exploiting compact representations of the defender's strategies. Unfortunately, we show this is not possible here. Interestingly, it turns out that the hardness of the problem depends on how many targets an attacker is interested in. In particular, we say that an attacker of type θ is *not interested* in attacking target i if there exists j such that $U^a_u(i|\theta) < U^a_c(j|\theta)$. That is, even when target i is totally uncovered and target j is fully covered, the attacker still prefers attacking target j – thus target i will never be attacked by an attacker of type θ. Otherwise we say that an attacker of type θ is *interested* in attacking target i. One might imagine that if an attacker is only interested in a small number of targets, this should simplify the computation. Interestingly, it turns out that this is *not* the case.

PROPOSITION 3.6. *Computing the optimal defender policy in a Bayesian Stackelberg security game (both with and without type-reporting IC constraints) is NP-hard, even when the defender payoff does not depend on the attacker's type and when each type of attacker is interested in attacking at most four targets.*

The proof of Proposition 3.6 requires a slight modification of a similar proof in [17], and is provided in the appendix just for completeness. Our next proposition shows that we are able to compute the optimal defender policy in a restricted setting. This setting is motivated by fare evasion deterrence [29] where each attacker (i.e., a passenger) is only interested in attacking (i.e., stealing a ride from) *one* specific target (i.e., the metro station nearby), or choosing to not attack (e.g., buying a ticket) in which case both players get utility 0. Formally, we model this as a setting where each attacker type is interested in two targets: one *type-specific* target and one *common* target $t_\emptyset$ (corresponding to the option of not attacking). If $t_\emptyset$ is attacked, each player gets utility 0 regardless of whether $t_\emptyset$ is protected or not – we call $t_\emptyset$ *coverage-invariant* for this reason.[5]

PROPOSITION 3.7. *Suppose each attacker type is interested in two targets: the* common coverage-invariant *target $t_\emptyset$ and a* type-specific *target. Then the defender's optimal policy (without type-reporting IC constraints) can be computed in poly$(m, n, |\Theta|)$ time.*

The proof of Proposition 3.7 crucially exploits the fact that each player's utility is "coverage-invariant" on target $t_\emptyset$. As a result, the defender will not cover $t_\emptyset$ at all at optimality. Therefore, for any attacker of type θ who is interested in target i and $t_\emptyset$, the defender only needs to signal information about the protection of target i. This allows us to write a linear program. The proof is deferred to the appendix. Note that when we take incentive compatibility constraints into account, the situation becomes more intricate. It could be the case that an attacker is not interested in attacking a target, but would still like to receive an informative signal regarding its coverage status in order to infer some information about the distribution of resources. This is reminiscent of information leakage as described by Xu et al. [27], and our proof does not naturally extend to this setting.

[5]The utility 0 is not essential so long as $t_\emptyset$ is coverage-invariant.

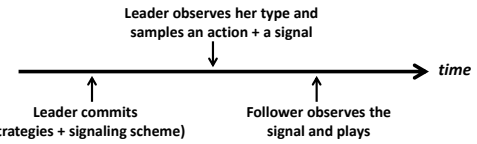


Figure 3: Timeline of the BSG With Multiple Leader Types

Interestingly, our next result shows that the restriction in Proposition 3.7 is almost necessary for efficient computation, as evidence of computational hardness manifests when we slightly go beyond the condition there.

PROPOSITION 3.8. *The defender oracle problem* [6] *is NP-hard (both with and without type-reporting IC constraints), even when each type of attacker is interested in two targets.*

4. MULTIPLE LEADER TYPES, SINGLE FOLLOWER TYPE

4.1 The Model

Similarly to Section 3, we still start with the normal-form Bayesian Stackelberg Game, but with multiple *leader* types and a single *follower* type. Following the notations in Section 3, an instance of such a BSG is also given by a set of tuples $\{(A^\theta, B^\theta, \lambda_\theta)\}_{\theta\in\Theta}$ where $A^\theta, B^\theta \in \mathbb{R}^{m\times n}$ are the payoff matrices of the leader (row player) and the follower (column player) respectively. However, Θ now is the set of leader types and λ_θ is the probability that the leader has type θ. Among its many applications, one key motivation of this model is from security domains. In security games, the follower, i.e., the attacker, usually does not have full information regarding the importance and vulnerability of the targets for attack, while the leader, i.e., the defender, possesses much better knowledge. This can be modeled as a BSG where the leader has multiple types and the single-type follower has a prior belief regarding the leader's types.

It is known that in this case, a set of linear programs suffices to compute the optimal mixed strategy to commit to [5]. We consider a richer model where the leader can additionally commit to a policy, namely a *signaling scheme*, of partially releasing her *type* and *action*. Formally, the leader commits to a mixed strategy $\mathbf{x}^\theta$ for each realized type θ and a signaling scheme φ which is a *stochastic* map from $\Theta \times [m]$ to Σ. We call the pair

$$(\{\mathbf{x}^\theta\}_{\theta\in\Theta}, \varphi) \text{ where } \mathbf{x}^\theta \in \Delta_m;\ \varphi : \Theta \times [m] \xrightarrow{rnd} \Sigma \qquad (4)$$

a *leader policy* in this setting. The game starts with the leader's commitment. Afterwards, the leader observes her own type, and then samples an action and a signal accordingly. The follower observes the signal and best responds. Figure 3 illustrates the timeline of the game.

4.2 Commitment to Optimal Leader Policy

Similarly to Observation 3.1, it is easy to see there exists an optimal leader policy with n signals where each signal recommends an action to the follower. Therefore, without loss of generality, we assume $\Sigma = \{\sigma_1, ..., \sigma_n\}$ where σ_j is a signal recommending action j to the follower.

[6]The optimal policy can be computed by an LP with exponential size. The defender oracle is essentially the dual of the LP. See the Appendix for a derivation of the defender oracle and proof of the hardness.

THEOREM 4.1. *The optimal* leader policy *defined in Formula* (4) *can be computed in* $poly(m, n, |\Theta|)$ *time by linear programming.*

PROOF. To represent the signaling scheme φ, let $\varphi(j|i,\theta)$ be the probability of sending signal σ_j, conditioned on the realized leader type θ and pure strategy i. Then $p^\theta_{ij} = \varphi(j|i,\theta)\cdot x^\theta(i)$ is the *joint* probability for the leader to take (pure) action i and send signal σ_j, conditioned on a realized leader type θ. The following linear program computes the optimal $\{p^\theta_{ij}\}_{i\in[m],j\in[n],\theta\in\Theta}$.[7]

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta}\lambda_\theta \sum_{ij} p^\theta_{ij} a^\theta_{ij} & \\ \text{subject to} & \sum_{i=1}^m\sum_{j=1}^n p^\theta_{ij} = 1, & \text{for } \theta\in\Theta. \\ & \sum_{i,\theta}\lambda_\theta p^\theta_{ij} b^\theta_{ij} \geq \sum_{i,\theta}\lambda_\theta p^\theta_{ij} b^\theta_{ij'}, & \text{for } j\neq j'. \\ & p^\theta_{ij}\geq 0, & \text{for all } i,j,\theta. \end{array} \qquad (5)$$

By letting $x^\theta(i) = \sum_{j=1}^n p^\theta_{ij}$ and $\varphi(j|i,\theta) = p^\theta_{ij}/x^\theta(i)$, we can recover the optimal defender policy $(\{\mathbf{x}^\theta\}_{\theta\in\Theta}, \varphi)$. □

4.3 Security Games

We now again consider the security game setting. We have shown in Section 3 that, when there are multiple follower types, the polynomial-time solvability of BSGs does not extend to even the simplest security game setting. Interestingly, it turns out that when the leader has multiple types, the optimal leader strategy and signaling scheme can be efficiently computed in fairly general settings, as we will show in this section.

Continuing the setup in Section 3.4, we first introduce a few more preliminaries. Note that θ is now the defender's type. In security games, any defender pure strategy, denoted as $\mathbf{e}$, is a subset of targets that are protected by this pure strategy. We will view $\mathbf{e}$ as a *binary* vector from $\{0,1\}^n$ with each entry specifying whether the corresponding target is protected or not in this pure strategy. Let $E = \{\mathbf{e}_1, ..., \mathbf{e}_L\}$ be the set of all pure strategies. Therefore, the *convex hull* of E

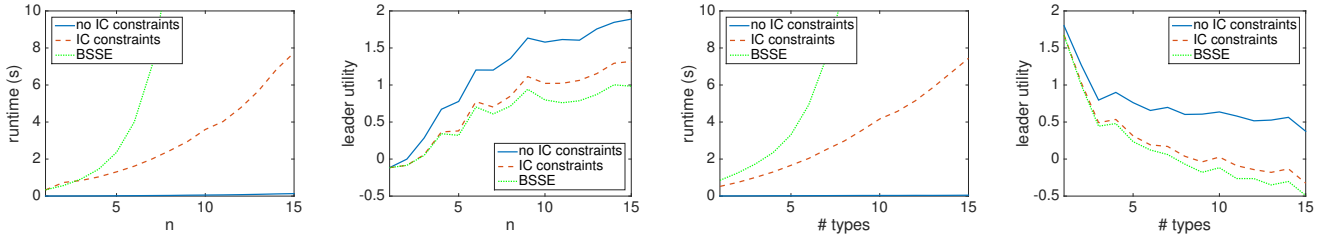
$$\mathcal{D} = Conv\{\mathbf{e}_1, ..., \mathbf{e}_L\} \qquad (6)$$

corresponds to the set of all *mixed strategies*, where a mixed strategy is summarized by the *marginal* coverage probabilities of each target. In security games, L is usually exponentially large in the natural representation, but $\mathcal{D}$ usually has compact representations, and moreover, both the defender's and attacker's utilities can be compactly represented using marginal probabilities. For example, with k identical unconstrained defending resources and n targets, $L = C^k_n = \mathcal{O}(n^k)$, the number of subsets of cardinality k, however $\mathcal{D}$ has a compact representation $\{\mathbf{x}\in\mathbb{R}^n : \sum_j x_j = k;\ x_j \in [0,1]\,\forall j\}$. But in many cases, security resources have scheduling constraints and $\mathcal{D}$ becomes more complicated. It can be shown that if the defender best response problem can be solved in polynomial time, then the Strong Stackelberg equilibrium can also be computed in polynomial time [13, 26]. We now establish an analogous result for BSG with signaling.

THEOREM 4.2. *The optimal defender policy can be computed in* $poly(n, |\Theta|)$ *time if the defender's best response problem (i.e., defender oracle) admits a* $poly(n)$ *time algorithm.*

PROOF. First, observe that LP (5) does not obviously extend to security game settings because the number of leader pure strategies

[7]Interestingly, when $|\Theta| = 1$, the game degenerates to a Stackelberg game without uncertainty of player types, and LP (5) degenerates to a linear program that computes the Strong Stackelberg equilibrium [4].



(a) Running time, $|\Theta|$ constant (b) Leader utility, $|\Theta|$ constant (c) Running time, n constant (d) Leader utility, n constant

Figure 4: Simulation results showing the effect of varying number of actions, n, and number of types, $|\theta|$, on the runtime and utility of the three different models in the case of multiple follower types.

is exponentially large here and so is the LP formulation. Therefore, like classic security game algorithms, it is crucial to exploit a compact representation of the leader's policy space. For this, we need an equivalent but slightly different view of the leader policy. That is, the leader policy can be *equivalently* viewed as follows: the leader observes her type θ and then randomly chooses a signal σ_j (occurring with probability $\sum_{i=1}^{m} p_{ij}^{\theta}$ in LP (5)), and finally picks a mixed strategy that depends on both θ and σ_j (i.e., the vector $(p_{1j}^{\theta}, p_{2j}^{\theta}, ..., p_{mj}^{\theta})$ normalized by the factor $\sum_{i=1}^{m} p_{ij}^{\theta}$ in LP (5)).

The different view of leader policy above allows us to write a quadratic program for computing the optimal leader policy. In particular, let p_j^{θ} be the probability that the leader sends signal j conditioned on the realized leader type θ, and let $\mathbf{x}_j^{\theta}$ be the leader's (marginal) mixed strategy conditioned on observing θ and sending signal σ_j. Then, upon receiving signal σ_j, a rational Bayesian attacker will updates his belief, and compute the expected utility for attacking target j' as

$$\sum_{\theta}\left(\frac{\lambda_\theta p_j^{\theta}}{\alpha_j}\cdot\left[\mathbf{x}_j^{\theta}(j')U_c^a(j'|\theta)+\left(1-\mathbf{x}_j^{\theta}(j')\right)U_u^a(j'|\theta)\right]\right) \quad (7)$$

where the normalization factor $\alpha_j = \sum_\theta \lambda_\theta p_j^{\theta}$ is the probability of sending signal σ_j. Define $AttU(j,j')$ to be the attacker utility by attacking target j' conditioned on receiving signal σ_j, multiplied by the probability α_j of receiving signal j. Formally,

$$\begin{aligned} & AttU(j,j') \\ = \;& \alpha_j \times \text{Equation (7)} \\ = \;& \textstyle\sum_\theta \left(\lambda_\theta p_j^{\theta}\mathbf{x}_j^{\theta}(j')U_c^a(j'|\theta)+\left[\lambda_\theta p_j^{\theta}-\lambda_\theta p_j^{\theta}\mathbf{x}_j^{\theta}(j')\right]U_u^a(j'|\theta)\right)\end{aligned}$$

Similarly, we can also define $DefU(j,j')$, the leader's expected utility of sending signal σ_j with target j' being attacked, scaled by the probability of sending σ_j. The attacker's incentive compatibility constraints are then $AttU(j,j) \geq AttU(j,j')$ for any $j' \neq j$. Then the leader's problem can be expressed as the following quadratic program with variables $\{\mathbf{x}_j^{\theta}\}_{j\in[n],\theta\in\Theta}$ and $\{p_j^{\theta}\}_{j\in[n],\theta\in\Theta}$.

$$\begin{array}{lll} \text{maximize} & \sum_j DefU(j,j) & \\ \text{subject to} & AttU(j,j) \geq AttU(j,j'), & \text{for } j \neq j'. \\ & \sum_j p_j^{\theta} = 1, & \text{for } \theta \in \Theta. \\ & \mathbf{x}_j^{\theta} \in \mathcal{D}, & \text{for } j,\theta. \\ & p_j^{\theta} \geq 0, & \text{for } j,\theta. \end{array} \quad (8)$$

The optimization program (8) is quadratic because $AttU(j,j')$ and $DefU(j,j')$ are quadratic in the variables. Notably, these two functions are *linear* in p_j^{θ} and the term $p_j^{\theta}\mathbf{x}_j^{\theta}$. Therefore, we define variables $\mathbf{y}_j^{\theta} = p_j^{\theta}\mathbf{x}_j^{\theta} \in \mathbb{R}^n$. Then, both $AttU(j,j')$ and $DefU(j,j')$ are linear in p_j^{θ} and $\mathbf{y}_j^{\theta}$. The only problematic constraint in program (8) is $\mathbf{x}_j^{\theta} \in \mathcal{D}$, which now becomes $\mathbf{y}_j^{\theta} \in p_j^{\theta}\mathcal{D}$ where both p_j^{θ} and $\mathbf{y}_j^{\theta}$ are variables. Here $p\mathcal{D}$ denotes the polytope $\{px : x \in \mathcal{D}\}$ for any given p. It turns out that this is still a convex constraint, and behaves nicely as long as the polytope $\mathcal{D}$ behaves nicely.

LEMMA 4.3. *Let $\mathcal{D} \subseteq \mathbb{R}^n$ be any* bounded *convex set. Then the following hold:*

(i) The extended set $\widetilde{\mathcal{D}} = \{(\mathbf{x}, p) : \mathbf{x} \in p\mathcal{D}, p \geq 0\}$ is convex.

(ii) If $\mathcal{D}$ is a polytope expressed by constraints $A\mathbf{x} \leq \mathbf{b}$, then $\widetilde{\mathcal{D}}$ is also a polytope, given by $\{(\mathbf{x}, p) : A\mathbf{x} \leq p\mathbf{b}, p \geq 0\}$;

(iii) If $\mathcal{D}$ admits a $poly(n)$ time separation oracle*, so does $\widetilde{\mathcal{D}}$.*[8]

The proof of Lemma 4.3 is standard, and is deferred to the appendix. We note that the restriction that $\mathcal{D}$ is bounded is important, otherwise some conclusions do not hold, e.g., Property 2. Fortunately, the polytope $\mathcal{D}$ of mixed strategies is bounded. Therefore, using Lemma 4.3, we can rewrite Quadratic Program (8) as the following linear program.

$$\begin{array}{lll} \text{maximize} & \sum_j DefU(j,j) & \\ \text{subject to} & AttU(j,j) \geq AttU(j,j'), & \text{for } j \neq j'. \\ & \sum_j p_j^{\theta} = 1, & \text{for } \theta \in \Theta. \\ & (\mathbf{y}_j^{\theta}, p_j^{\theta}) \in \widetilde{\mathcal{D}}, & \text{for } j,\theta. \\ & p_j^{\theta} \geq 0, & \text{for } j,\theta. \end{array} \quad (9)$$

Program (9) is linear because $AttU(j,j')$ and $DefU(j,j)$ are linear in p_j^{θ} and $\mathbf{y}_j^{\theta}$, and moreover, $(\mathbf{y}_j^{\theta}, p_j^{\theta}) \in \widetilde{\mathcal{D}}$ are essentially linear constraints due to Lemma 4.3 and the fact that $\mathcal{D}$ is a polytope in security games. Furthermore, LP (9) has a compact representation as long as the polytope of realizable mixed strategies $\mathcal{D}$ has one. In this case, LP (9) can be solved explicitly. More generally, by standard techniques from convex programing, we can show that the separation oracle for $\mathcal{D}$ easily reduces to the defender's best response problem. Thus if the defender oracle admits a $poly(n)$ time algorithm, then a separation oracle for $\mathcal{D}$ can be found in $poly(n)$ time. By Lemma 4.3, $\widetilde{\mathcal{D}}$ then admits a $poly(n)$ time separation oracle, so LP (9) can solved in $poly(n, |\Theta|)$ time. The proof is not particularly insightful and a similar argument can be found in [26]. So we omit the details here. □

4.4 Relation to Other Models

We note that our model in this section is related to several models from the literature on both information economics and security games. In particular, when the leader does not have actions

[8] A separation oracle for a convex set $\mathcal{D} \subseteq \mathbb{R}^n$ is an algorithm, which, given any $\mathbf{x_0} \in \mathbb{R}^n$, either correctly asserts $\mathbf{x_0} \in \mathcal{D}$ or asserts $\mathbf{x_0} \notin \mathcal{D}$ and find a hyperplane $\mathbf{a}\cdot\mathbf{x} = b$ separating x_0 from $\mathcal{D}$ in the following sense: $\mathbf{a}\cdot\mathbf{x_0} > b$ but $\mathbf{a}\cdot\mathbf{x} \leq b$ for any $\mathbf{x} \in \mathcal{D}$. It is well-known that the convex program $\max\ \mathbf{a}\cdot\mathbf{x}$ subject to $\mathbf{x} \in \mathcal{D}$ can be solved in $poly(n)$ time for any $\mathbf{a} \in \mathbb{R}^n$ if $\mathcal{D}$ has a $poly(n)$ time separation oracle [11].

and only privately observes her type, our model degenerates to the *Bayesian Persuasion* (BP) model of [15]. The BP model is a two-player game played between a *sender* (leader in our case) and a *receiver* (follower in our case). The receiver must take one of a number of actions with a-priori *unknown* payoff, and the sender has *no* actions but possesses additional information regarding the payoff of various receiver actions (i.e., the leader observes her type). The BP model studies how the sender can signal her additional information to persuade the receiver to take an action that is more favorable to the sender. Variants of the BP model have been applied to varied domains including auctions, advertising, voting, multi-armed bandits, medical research and financial regulation. For additional references, see [7]. Our model generalizes the BP model to the case where sender has both actions and additional private information, and our results show that this generalized model can be solved in fairly general settings.

The security game setting in this section also relates to the model of Rabinovich et al. [23]. Rabinovich et al. considered a similar security setting where the defender can partially signal her strategy and extra knowledge about targets' states to the attacker in order to achieve better defender utility. This is essentially a BSG with multiple leader types and a single follower type. Rabinovich et al. [23] were able to efficiently solve for the case with unconstrained identical security resources. Our Theorem 4.2 shows that this model can actually be efficiently solved in much more general security settings allowing complicated real-world scheduling constraints, as long as the defender oracle problem can be solved efficiently.

5. SIMULATIONS

We will mainly present the comparison of the models discussed in Section 3 in terms of both the leader's optimal utility and the runtime required to compute the leader's optimal policy. We focus primarily on the setting with one leader type and multiple follower types, for two reasons. First, this is the case in which it is NP-hard to compute the optimal leader strategy without allowing the leader to signal (i.e., to compute the BSSE strategy), while our models of signaling permit a polynomial time solution. Second, some interesting phenomena in our simulations for the case of multiple leader types also show up in the case of multiple follower types.

We generate random instances using a modification of the covariant game model [19]. In particular, for given values of m, n, and Θ, we independently set a_{ij}^{θ} equal to a random integer in the range $[-5, 5]$ for each i, j, θ. Probabilities $\{\lambda_\theta\}_{\theta\in\Theta}$ were generated randomly. For some value of $\alpha \in [0, 1]$, we then set $B = \alpha(B') + (1-\alpha)(-A)$, where B' is a random matrix generated in the same fashion as A. So in the case that $\alpha = 0$ the game is zero-sum, while $\alpha = 1$ means independent and uniform random leader and follower payoffs. For every set of parameter values, we averaged over 50 instances generated in this manner to obtain the utility/runtime values we report.

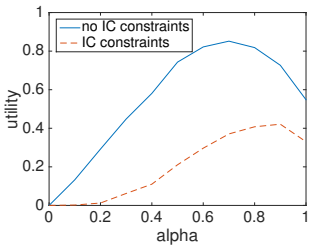


Figure 5: Extra utility gained by the leader from signaling.

We first consider the value of signaling for different values of α chosen from the set $\{0, 0.1, 0.2, ..., 1\}$. For these simulations, we fixed $m = n = 10$ and $|\Theta| = 5$. Figure 5 shows the *absolute* increase in leader utility from signaling (both with and without the type-reporting IC constraints), compared with the utility from BSSE (the $y = 0$ baseline). Note that when $\alpha = 0$ there is no gain from signaling, from Proposition 3.3. Interestingly, the gain from signaling is non-monotone, peaking at around $\alpha = 0.7$. Intuitively, large α means low correlation between the payoff matrices of the leader and follower, therefore there is a high probability that some entries will induce high payoff to both players. The leader can therefore extract high utility from commitment alone, thus derives little gain from signaling. However, as we decrease α and the game becomes more competitive, commitment alone is not as powerful for the leader and she has more to gain from being able to signal.

We next investigate the relation between the size of the BSG and the leader's utility, as well as runtime, for the three different models. In Figures 4(a) and 4(b), we hold the number of follower types constant ($|\Theta| = 5$) and vary $m = n$ between 1 and 15. In Figures 4(c) and 4(d) we fix $m = n = 5$ and vary $|\Theta|$ between 1 and 15. In all cases we set $\alpha = 0.5$ for generating random instances.

Not surprisingly, allowing signaling (both with and without the IC constraints) provides a significant speed-up over computing the BSSE.[9] On the other hand, the additional constraints in the model with IC constraints also increase the running time over the model without those constraints. Indeed, the time to compute the leader's optimal policy without the IC constraints appears as a flat line in Figures 4(a) and 4(c).

In both figures of leader utility, the differences of the leader's utility among the models are as indicated by Proposition 3.5. Observe that in all models the leader's utility increases with the number of actions, but decreases with the number of types. One explanation is that the former effect is due to the increased probability that the payoff matrices for a given follower type contain 'cooperative' entries where both players achieve high utility. However, as the number of follower types increases, it becomes less likely that the leader's strategy (which does not depend on the follower type) can "cooperate" with a majority of follower types simultaneously. Thus there is an increased chance that the leader's strategy results in low utilities when playing against a reasonable fraction of follower types, which accounts for the latter effect.

In the case of multiple leader types, allowing the leader to signal actually results in a small computational speed up compared to the case without signaling. We hypothesize that this is because we only need to solve one LP to compute the optimal policy, rather than the multiple LPs required to solve without signaling [5]. Unsurprisingly, we also see an increase in the leader's utility. The utility trends are similar to the case of multiple follower types, so we do not present them in detail.

6. CONCLUSIONS AND DISCUSSIONS

In this paper, we studied the effect of signaling in Bayesian Stackelberg games. We show that the leader's power of commitment to a signaling scheme not only achieves higher utility, but also computational speed-ups. Some of the polynomial-time solvability results extend to security games, an important application domain of Stackelberg games, while others cease to hold. There are many interesting directions for future work. What if different follower types can share information with each other? For a Bayesian leader, what if her signaling scheme cannot be correlated with her mixed strategy, but only carries information about her type? Can we apply these ideas to other domains, e.g., mechanism design where the mechanism designer implicitly serves as the leader?

[9]To compute the BSSE, we implement the state-of-art algorithm DOBBS, a mixed integer linear program as formulated in [21].

Acknowledgments: This research is supported by NSF grant CCF-1350900 and MURI grant W911NF-11-1-0332. Part of the research is done when the authors are visiting the Simons Institute for the Theory of Computing.

REFERENCES

[1] T. M. Anton Kolotilin, Ming Li and A. Zapechelnyuk. Persuasion of a privately informed receiver. *Working Paper*, 2015.

[2] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[3] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash Equilibria. *J. ACM*, 56(3):14:1–14:57, May 2009.

[4] V. Conitzer and D. Korzhyk. Commitment to correlated strategies. In *Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI)*, 2011.

[5] V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pages 82–90. ACM, 2006.

[6] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash Equilibrium. In *Proceedings of the Thirty-eighth Annual ACM Symposium on Theory of Computing*, STOC '06, pages 71–78, New York, NY, USA, 2006. ACM.

[7] S. Dughmi and H. Xu. Algorithmic Bayesian persuasion. In *Proceedings of the Forty-eighth Annual ACM Symposium on Theory of Computing*, STOC '16, 2016.

[8] Y. Emek, M. Feldman, I. Gamzu, R. Paes Leme, and M. Tennenholtz. Signaling schemes for revenue maximization. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pages 514–531, New York, NY, USA, 2012. ACM.

[9] F. Etro. Stackelberg, heinrich von: Market structure and equilibrium. *Journal of Economics*, 109(1):89–92, 2013.

[10] F. Fang, P. Stone, and M. Tambe. When security games go green: Designing defender strategies to prevent poaching and illegal fishing. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2015.

[11] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, 1988.

[12] M. Guo and A. Deligkas. Revenue maximization via hiding item attributes. *CoRR*, abs/1302.5332, 2013.

[13] M. Jain, E. Kardes, C. Kiekintveld, F. Ordóñez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In M. Fox and D. Poole, editors, *Proceedings of the 24th AAAI Conference on Artificial Intelligence (AAAI)*. AAAI Press, 2010.

[14] A. X. Jiang and K. Leyton-Brown. Polynomial-time computation of exact correlated equilibrium in compact games. In *Proceedings of the Twelfth ACM Electronic Commerce Conference (ACM-EC)*, 2011.

[15] E. Kamenica and M. Gentzkow. Bayesian persuasion. *American Economic Review*, 101(6):2590–2615, 2011.

[16] J. Letchford, V. Conitzer, and K. Munagala. Learning and approximating the optimal strategy to commit to. In M. Mavronicolas and V. G. Papadopoulou, editors, *SAGT*, volume 5814 of *Lecture Notes in Computer Science*, pages 250–262. Springer, 2009.

[17] Y. Li, V. Conitzer, and D. Korzhyk. Catcher-evader games. *arXiv:1602.01896*.

[18] P. B. Miltersen and O. Sheffet. Send mixed signals: earn more, work less. In B. Faltings, K. Leyton-Brown, and P. Ipeirotis, editors, *EC*, pages 234–247. ACM, 2012.

[19] E. Nudelman, J. Wortman, Y. Shoham, and K. Leyton-Brown. Run the GAMUT: A comprehensive approach to evaluating game-theoretic algorithms. In *Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems-Volume 2*, pages 880–887. IEEE Computer Society, 2004.

[20] C. H. Papadimitriou and T. Roughgarden. Computing correlated equilibria in multi-player games. *J. ACM*, 55(3):14:1–14:29, Aug. 2008.

[21] P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Efficient algorithms to solve Bayesian Stackelberg games for security applications. In *Proceedings of the 23rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1559–1562, 2008.

[22] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed armor protection: the application of a game theoretic model for security at the Los Angeles international airport. In *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems: industrial track*, pages 125–132. International Foundation for Autonomous Agents and Multiagent Systems, 2008.

[23] Z. Rabinovich, A. X. Jiang, M. Jain, and H. Xu. Information disclosure as a means to security. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2015, Istanbul, Turkey, May 4-8, 2015*, pages 645–653, 2015.

[24] L. Rayo and I. Segal. Optimal information disclosure. *Journal of Political Economy*, 118(5):949 – 987, 2010.

[25] M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, New York, NY, USA, 1st edition, 2011.

[26] H. Xu, F. Fang, A. X. Jiang, V. Conitzer, S. Dughmi, and M. Tambe. Solving zero-sum security games in discretized spatio-temporal domains. In *Proceedings of the 28th Conference on Artificial Intelligence (AAAI 2014), Québec, Canada*, 2014.

[27] H. Xu, A. X. Jiang, A. Sinha, Z. Rabinovich, S. Dughmi, and M. Tambe. Security games with information leakage: Modeling and computation. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 674–680, 2015.

[28] H. Xu, Z. Rabinovich, S. Dughmi, and M. Tambe. Exploring information asymmetry in two-stage security games. In *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI)*, 2015.

[29] Z. Yin, A. Jiang, M. Johnson, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems. In *Proceedings of the Conference on Innovative Applications of Artificial Intelligence (IAAI)*, 2012.

APPENDIX

A. PROOF OF PROPOSITION 3.3

First, notice that $U_{sig}(G) \geq U_{BSSE}(G)$ for any BSG G (not necessarily zero-sum). This is because the leader policy of playing the BSSE leader mixed strategy and sending only *one* signal to each attacker type degenerates to the BSSE. We now show that $U_{sig}(G) \leq U_{BSSE}(G)$. Let $(\mathbf{x}^*, p)$ be the optimal leader policy computed by LP (2). Note that, if the leader plays the optimal leader policy $(\mathbf{x}^*, p)$, but the follower type θ "irrationally" ignores any signal and simply reacts to $\mathbf{x}^*$ by taking the best response (to $\mathbf{x}^*$) action j^*, then, the follower of type θ gets utility $\sum_i x_i^* b_{ij^*}^\theta$. We claim that this utility is less than the utility of best responding to each signal separately, as shown below

$$\sum_j \sum_i p_{ij}^\theta b_{ij}^\theta \geq \sum_j \sum_i p_{ij}^\theta b_{ij^*}^\theta = \sum_i x_i^* b_{ij^*}^\theta$$

where the inequality is due to second set of constraints in LP (2) and the equality is due to the first set of constraints in LP (2). Since this is a zero-sum game, the leader will be better off if the follower of type θ ignores signals. Let U be the defender utility when all the attacker types best respond to $\mathbf{x}^*$ by ignoring signals, then $U \geq U_{sig}(G)$. However, U is simply the defender utility in this BSG by committing to the mixed strategy $\mathbf{x}^*$ without any signaling, therefore is upper bounded by $U_{BSSE}(G)$. As a result, $U_{BSSE}(G) \geq U \geq U_{sig}(G)$, as desired.

B. PROOF OF PROPOSITION 3.6

This is a slight modification from a proof of the hardness of Bayesian Stackelberg games (Theorem 2 in [17]). We provide it only for completeness.

The reduction is from 3-SAT. Given an instance of 3-SAT with n variables and m clauses, we create a security game with $2n + 2$ targets and n resources. For each variable, there is a target corresponding to taht variable and its negation (call these *variable* targets), as well as a *punishment* and a *reward* target.

There are $m + 3n$ types of attacker. m of these are *clause* types, one per clause. Each of these types are interested in attacking all targets corresponding to literals appearing in the corresponding clause, or the reward target. For any literal contained in the clause, this type gets -1 payoff for attacking when the target is covered and 0 when it is uncovered. Any clause type attacker gets 0 payoff for attacking the punishment target, whether or not it is covered. Note that if a clause type believes that at least one of the literal targets is covered with probability 1, then they will attack that target (breaking ties favorably). Otherwise, they attack the punishment target.

There is one pair type for each variable. These types are not interested in any literal target that does not correspond to the relevant variable, or the reward target. For the two literal targets they are interested in, they get -1 payoff for attacking a covered target and 0 for an uncovered target. They get 0 for attacking the punishment target. Again, a pair type target will only not attack the punishment target if they believe that both literal targets are covered with non-zero probability.

Lastly there are $2n$ counting types, one per literal. Each of these types is not interested in any literal target other than the one corresponding to them, or the punishment node. If they attack the relevant literal node and it is covered they get 0 payoff, and if it is uncovered they get 1. They get 0 payoff for attacking the reward target, regardless of whether it is covered. Note that each of these types attacks the reward target if they believe that the literal target is covered with probability 1.

The defender gets 0 payoff whenever a literal target is attacked, regardless of whether it is covered and -1 payoff whenever the punishment target is attacked. If any attacker attacks the reward target the defender gets payoff (note that the only attacker types that will ever attack the reward target are the counting types).

Each type occurs with equal probability.

We show that the defender can obtain a utility of $\frac{n}{m+3n}$ if and only if the instance of 3-SAT is satisfiable.

If the instance is satisfiable, then we simply cover the variable targets corresponding to a satisfying assignment, and signal as such. Then all clauses are satisfied, so no clause type attacks the punishment node, no variable has both its positive and negative literals covered with positive probability, and n counting types are sure that their literal is covered, so they attack the reward node. This results in an expected utility of $\frac{n}{m+3n}$ for the defender.

Now suppose the instance of 3-SAT is not satisfiable. Note that whenever there is any uncertainty for the attacker they take an undesirable action, therefore the defender optimally signals truthfully about their chosen action. Since the instance is unsatisfiable, for any allocation of resources either a clause type or pair type will be incentivized to attack the punishment target. The defender can get payoff 1 at most $\frac{n}{m+3n}$ of the time (from exactly n counting types, as the defender can cover only n variable targets at a time), and gets -1 payoff from the pair/clause type that attacks the punishment target. Therefore the defender gets less than $\frac{n}{m+3n}$ expected utility.

C. PROOF OF PROPOSITION 3.7

For convenience, let target 0 denote the common coverage-invariant target. By assumption, let i_θ denote the only type-specific target for the attacker of type θ. Notice that, our signaling scheme only needs two signals for the attacker of type θ, recommending either target i_θ or target 0 for attack, since he is not interested in other targets. Therefore, for each attacker type θ, we define four variables: $p_{c,j}^\theta$ [$p_{u,j}^\theta$] is the probability that type θ's specific target i_θ is covered [uncovered] and action j is recommended to the attacker, where $j \in \{i_\theta, 0\}$ is either to attack i_θ, or stay home. Notice that, we can define these variables because our signaling scheme for type θ only depends on the coverage status of target i_θ as the utility of the common target 0 is coverage-invariant. This is crucial, since otherwise, the optimal signaling scheme may depend on all the targets that type θ is interested, and this makes the problem much harder (as shown in Proposition 3.8). The following linear program, with variables $p_{c,j}^\theta$ and $\mathbf{x}$, computes the optimal defender utility.

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta} \lambda_\theta \sum_{s\in\{c,u\}} p_{s,i_\theta}^\theta U_x^d(i_\theta;\theta) & \\ \text{subject to} & \sum_{j\in\{0,i_\theta\}} p_{c,j}^\theta = x_{i_\theta}, & \text{for } \theta \in \Theta. \\ & \sum_{j\in\{0,i_\theta\}} p_{u,j}^\theta = 1 - x_{i_\theta}, & \text{for } \theta \in \Theta. \\ & \sum_{s\in\{c,u\}} p_{s,j}^\theta U_s^a(j;\theta) \geq & \\ & \quad \sum_{s\in\{c,u\}} p_{s,j}^\theta U_s^a(j';\theta), & \text{for } \theta \in \Theta. \\ & \mathbf{x} \in \mathcal{D} & \end{array} \tag{10}$$

where: the first two constraints mean that the signaling scheme should be consistent with the true marginal probability that i is covered (first constraint) or uncovered (second constraint). The third constraint is the incentive compatibility constraint which guarantees that the attacker prefers to follow the recommended action. The last constraint ensures that the marginal distribution $\mathbf{x}$ is implementable ($\mathcal{D}$ is the set of all implementable marginals. See Section 4.3 for more information.)

D. PROOF OF PROPOSITION 3.8

D.1 LP Formulation of the Problem and its Dual

Using similar notations as Section 4.3, we equivalently regard each pure strategy as a vector $e \in \{0,1\}^n$, and E is the set of all pure strategies. We consider the case where the defender does not have any scheduling constraints, i.e., e is any vector with at most k 1's, and show that the defender oracle in this basic setting is already NP-hard. To describe a mixed strategy, let p_e be the probability of taking pure strategy e. Then

$$x = \mathbb{E}(e) = \sum_{e \in \mathcal{S}} e \times p_e \tag{11}$$

is the marginal coverage probability corresponding to this pure strategy $\{p_e\}_{e\in\mathcal{S}}$. Notice that $x \in R^n$.

By Observation 3.1, n signals are need for each attacker type in the optimal scheme. Therefore, let $p^\theta_{s,i}$ be the probability that pure strategy s is taken and the attacker of type θ is recommended to take action i. Then $\alpha^\theta_i = \sum_{e\in E} p^\theta_{e,i}$ is the probability that attacker of type θ is recommended to take action i, while

$$x^\theta_i = \sum_{e\in E} e \times p^\theta_{e,i}$$

is the corresponding posterior belief (absent by a normalization factor $1/\alpha^\theta_i$) of marginal coverage when the attacker of type θ is recommended action i. Then the following optimization formulation computes the defender's optimal mixed strategy as well as signaling scheme.[10]

$$\begin{array}{lll}
\text{maximize} & \sum_{\theta,i} \lambda_\theta \left[x^\theta_{ii} U^c_d(i;\theta) + (\alpha^\theta_i - x^\theta_{ii}) U^u_d(i;\theta)\right] & \\
\text{subject to} & x^\theta_{ii} U^c_a(i,\theta) + (\alpha^\theta_i - x^\theta_{ii}) U^u_a(i,\theta) \geq & \\
& \quad x^\theta_{ij} U^c_a(j,\theta) + (\alpha^\theta_i - x^\theta_{ij}) U^u_a(j,\theta), & \text{for } i,j,\theta. \\
& \alpha^\theta_i = \sum_{e\in E} p^\theta_{e,i}, & \text{for } i,\theta. \\
& \sum_{e\in E} e \times p^\theta_{e,i} = x^\theta_i, & \text{for } i,\theta. \\
& \sum_{i=1}^n p^\theta_{e,i} = p_e, & \text{for } e,\theta. \\
& \sum_{s\in E} p_e = 1 & \\
& p^\theta_{e,i} \geq 0, p_e \geq 0, & \text{for } e,i,\theta.
\end{array} \tag{12}$$

where $x^\theta_i \in R^n, p_s \in \mathbb{R}, p^\theta_{s,i} \in \mathbb{R}$ are variables.

We now take the dual of LP (12). Instead of providing the exact dual program, we abstractly represent the dual by highlighting the non-trivial part, as follows:

$$\begin{array}{lll}
\text{minimize} & \gamma & \\
\text{subject to} & poly(n, |\Theta|) \text{ linear constraints on } y^\theta_i, \beta^\theta_i & \\
& -\beta^\theta_i + e \cdot y^\theta_i + q^\theta_e \geq 0, & \text{for } i,e,\theta. \\
& \sum_\theta -q^\theta_e + \gamma \geq 0, & \text{for } e.
\end{array} \tag{13}$$

where $\beta^\theta_i, q^\theta_e, \gamma \in \mathbb{R}, y^\theta_i \in \mathbb{R}^n$ are variables. We now analyze the dual program (13). Notice that the first (implicitly described) constraint does not depend on γ, q^θ_e. So the last constraint, together with the "min" objective, yields that $\gamma = \max_{e\in E} \sum_\theta q^\theta_e$ at optimality. The middle constraint, together with the "min" objective, yields that $q^\theta_e = \max_i[\beta^\theta_i - e \cdot y^\theta_i]$ at optimality. As a result, the dual program can be re-written in the following form:

$$\max_{e\in E} \left[\sum_\theta \max_i (\beta^\theta_i - e \cdot y^\theta_i)\right]$$
$$s.t. \qquad poly(n, |\Theta|) \text{ linear constraints on } y^\theta_i, \beta^\theta_i.$$

[10] We only consider the case with no IC constraints for incentivizing attacker's *type report*. Adding IC constraint will result in the same defender oracle, thus is omitted here.

Notice that, this is still a convex program – the objective can be viewed as maximizing a convex function.

D.2 The Defender Oracle

The defender oracle problem is precisely to evaluate the function

$$f(y^\theta_i, \beta^\theta_i) = \max_{e\in E} \left[\sum_\theta \max_i (\beta^\theta_i - e \cdot y^\theta_i)\right] \tag{14}$$

for any given input $y^\theta_i, \beta^\theta_i$. When the attacker of type θ is only interested in a small number of targets, say a subset S of targets. Then in LP (12), the third constraint on $x^\theta_i \in \mathbb{R}^n$ only needs to be restricted to the targets in S, since the attacker of type θ does not care about the coverage of other targets at all. That is, there is no constraints for x^θ_i for all $i \notin S$; Moreover, for those $i \in S$, the constraint on x^θ_i can be restricted to only the entries in S. This simplification is reflected in the defender oracle problem in the following way: the input y^θ_i are non-zeros vectors only for those $i \in S$; moreover, the non-zero y^θ_i only has non-zeros at those entries corresponding to S.

D.3 Hardness of the Defender Oracle

We now prove that the defender oracle problem is NP-hard, even when each attacker type θ is only interested in 2 targets. In other words, we prove that evaluating function $f(y^\theta_i, \beta^\theta_i)$ is NP-hard, even when only two y^θ_i's are non-zero vectors for each θ and each of these two y^θ_i's only has two non-zero entries.

We reduce from max-cut. Given any graph $G = (V, \Theta)$ with node set V and edge set Θ. Construct a security game with V as targets and Θ as attacker types. The attacker type $\theta = (i,j)$ is interested in only targets i, j. For any type $\theta = (i,j)$, define y^θ_i as follows: $y^\theta_{ii} = 1$, $y^\theta_{ij} = -1$ and $y^\theta_{ik} = 0$ for any $k \neq i,j$; define y^e_j as follows: $y^\theta_{ji} = -1$, $y^\theta_{jj} = 1$ and $y^\theta_{jk} = 0$ for any $k \neq i,j$. Let $\beta^\theta_i = 0$ for any i, θ. We will think of each pure strategy e as a cut of size k, with all value-1 nodes on one side and value-0 nodes on another side. Let

$$c(e) = \sum_{\theta\in\Theta} \max_k (\beta^\theta_k - e \cdot y^\theta_k) = \sum_{\theta=(i,j)\in\Theta} \max(-e \cdot y^\theta_i, -e \cdot y^\theta_j).$$

Note that $\max(-e \cdot y^\theta_i, -e \cdot y^\theta_j) = 1$ if and only if edge θ is cut by strategy e (in which case $e \cdot y^\theta_i, e \cdot y^\theta_j$ equals $1, -1$ respectively). Otherwise $\max(-e \cdot y^\theta_i, -e \cdot y^\theta_j) = 0$. Therefore, $c(e)$ equals precisely the cut size induced by e. Note that evaluating function f defined in Equation (14) is to maximize $c(e)$ over $e \in E$, which is precisely to compute the Max k-Cut, a well-known NP-hard problem. Therefore the defender oracle is NP-hard, even when each attacker type is only interested in two targets.

E. PROOF OF LEMMA 4.3

Part 1: This is standard, and can be found, e.g., in [2]. We provide a proof for completeness. Consider any two elements $(\mathbf{x}, p)$ and $(\mathbf{y}, q)$ from $\widetilde{\mathcal{D}}$. So there exists $\mathbf{a}, \mathbf{b} \in \mathcal{D}$ such that $\mathbf{x} = p \cdot \mathbf{a}$ and $\mathbf{y} = q \cdot \mathbf{b}$. To prove the convexity, we need to show $\alpha \cdot (\mathbf{x}, p) + \beta \cdot (\mathbf{y}, q) \in \widetilde{\mathcal{D}}$ for any $\alpha \in (0,1)$ and $\alpha + \beta = 1$. If $p = q = 0$, this is obvious; Otherwise, we have

$$\begin{aligned}
\alpha \cdot (\mathbf{x}, p) + \beta \cdot (\mathbf{y}, q) &= \alpha(p \cdot \mathbf{a}, p) + \beta(q \cdot \mathbf{b}, q) \\
&= \left([\alpha p + \beta q] \cdot \frac{\alpha p \cdot \mathbf{a} + \beta q \cdot \mathbf{b}}{\alpha p + \beta q}, \alpha p + \beta q\right)
\end{aligned}$$

Notice that $\frac{\alpha p \cdot \mathbf{a} + \beta q \cdot \mathbf{b}}{\alpha p + \beta q} \in \mathcal{D}$ due to the convexity of $\mathcal{D}$, therefore $\alpha \cdot (\mathbf{x}, p) + \beta \cdot (\mathbf{y}, q) \in \widetilde{\mathcal{D}}$. So $\widetilde{\mathcal{D}}$ is convex.

Part 2: First, it is easy to see that any element from $\widetilde{\mathcal{D}}$ satisfies $A\mathbf{x} \leq p\mathbf{b}$ and $p \geq 0$. We prove the other direction. Namely, for any $(\mathbf{x}, p)$ satisfies $A\mathbf{x} \leq p\mathbf{b}$ and $p \geq 0$, $(\mathbf{x}, p) \in \widetilde{\mathcal{D}}$. It is easy to see that this is true for $p > 0$ since $x/p \in \mathcal{D}$. The non-trivial part is when $p = 0$, in which case $(\mathbf{x}, p) \in \widetilde{\mathcal{D}}$ if and only if $\mathbf{x} = 0$. We need to prove the only $\mathbf{x}$ satisfying $A\mathbf{x} \leq 0$ is the all-zero vector $\mathbf{0}$. Here we need the condition that $\mathcal{D}$ is bounded. If (by contradiction) there exists $\mathbf{x}_0 \neq \mathbf{0}$ satisfying $A\mathbf{x}_0 \leq 0$, then for any $\mathbf{x} \in \mathcal{D}$, we must have $\mathbf{x} + \alpha\mathbf{x}_0 \in \mathcal{D}$ for any $\alpha > 0$, which contradicts the fact that $\mathcal{D}$ is bounded.

Part 3: If $\mathcal{D}$ has a separation oracle $\mathcal{O}$, then the following is a separation oracle for $\widetilde{\mathcal{D}}$. Given arbitrary $(\mathbf{x}_0, p_0) \in \mathbb{R}^{n+1}$,

case 1: If $p_0 < 0$, return "no" and separation hyperplane $p_0 = 0$;

case 2: If $p_0 > 0$, first check whether $\mathbf{x}_0/p_0 \in \mathcal{D}$. If this is true, return "yes"; otherwise, find a violated constraint, using oracle $\mathcal{O}$, such that $\mathbf{a}^T \cdot \frac{\mathbf{x}_0}{p_0} > b$ but $\mathbf{a}^T \cdot \mathbf{x}' \leq b$ for any $\mathbf{x}' \in \mathcal{D}$. We claim that $\mathbf{a}^T \cdot \mathbf{x} - bp = 0$ is a hyperplane separating $(\mathbf{x}_0, p_0)$ from $\widetilde{\mathcal{D}}$. In particular, for any $(\mathbf{x}, p) \in \widetilde{\mathcal{D}}$ with $p > 0$, $\exists \mathbf{x}' \in \mathcal{D}$ such that $\mathbf{x}/p = \mathbf{x}'$. Note that $\mathbf{a}^T \cdot \mathbf{x}' \leq b$ since $\mathbf{x}' \in \mathcal{D}$, so $\mathbf{a}^T \cdot \mathbf{x} \leq pb$ (also holds when $p = 0$ in which case $\mathbf{x} = \mathbf{0}$) . However $\mathbf{a}^T \cdot \mathbf{x}_0 > p_0 b$. Therefore, $\mathbf{a}^T \cdot \mathbf{x} - pb = 0$ is a separation hyperplane.

case 3: If $p_0 = 0$, return "yes" if $\mathbf{x}_0 = \mathbf{0}$. Otherwise, return "no", and find a separation hyperplane as follows. Since $\mathcal{D}$ is bounded, we can find some $L_0 > 0$ large enough such that $\mathbf{y}_0 = L_0\mathbf{x}_0 \notin conv(\mathcal{D}, 0)$, where $conv(\mathcal{D}, 0)$ is the convex hull of $\mathcal{D}$ and the origin 0 (thus contains $\mathcal{D}$), and is introduced for technical convenience. Let $\mathbf{a} \cdot \mathbf{y} = b$ be a hyperplane separating $\mathbf{y}_0$ from $conv(\mathcal{D}, 0)$. That is $\mathbf{a} \cdot \mathbf{y}_0 > b$ and $\mathbf{a} \cdot \mathbf{y} \leq b$ for any $\mathbf{y} \in conv(\mathcal{D}, 0)$, in particular, for any $\mathbf{y} \in \mathcal{D}$. Similarly to the argument in case 2, we know that $\mathbf{a} \cdot \mathbf{x} \leq pb$ for any $(\mathbf{x}, p) \in \widetilde{\mathcal{D}}$. Note that, since $0 \in conv(\mathcal{D}, 0)$, we have $b \geq \mathbf{a} \cdot 0 = 0$ is non-negative. As a result, $\mathbf{a} \cdot L\mathbf{x}_0 = \frac{L}{L_0}\mathbf{a} \cdot \mathbf{y}_0 > b$ for any $L \geq L_0$. That is, $\mathbf{a} \cdot \mathbf{x}_0 > \frac{1}{L}b$ for any $L \geq L_0$. Therefore, we must have $\mathbf{a} \cdot \mathbf{x}_0 \geq 0 = p_0 b$ since $p_0 = 0$. As a result, the hyperplane $\mathbf{a} \cdot \mathbf{x} = pb$ separates $(\mathbf{x}_0, p_0)$ from $\widetilde{\mathcal{D}}$.