Information as a Double-Edged Sword in Strategic Interactions

by

Haifeng Xu

A Dissertation Presented to the
FACULTY OF THE GRADUATE SCHOOL
UNIVERSITY OF SOUTHERN CALIFORNIA
In Partial Fulfillment of the
Requirements for the Degree
DOCTOR OF PHILOSOPHY
(Computer Science)

May 2019

Acknowledgment

First and foremost, I would like to thank my advisors, Shaddin Dughmi and Milind Tambe,[1] whose careful mentorship and tremendous support largely influenced my growth as an academic. Being co-supervised is a very unique experience; Shaddin and Milind have made this a really harmonious and rewarding journey for me. I am very grateful for getting exposed to different research fields and experiencing different supervision styles, which also helped to shape my own perspectives. I also thank them for providing me the freedom to work on problems of my own interest, but always being available to help whenever I needed them. Over the years, their guidance has been a constant source of inspiration and helped me to develop as an independent researcher. I learned a lot from Shaddin through his beautiful taste of research, deep insights, fast thinking, clarity of thought and explanation, artistic paper writing, and the list goes on. Milind taught me what are important things to do at different stages, how to find significant research problems to work on, and how to explain research to any group of audience. I am also grateful to Milind for providing me the opportunity to work on cool real-world projects, such as FAMS and PAWS.

I would like to thank my committee members: Odilon Câmara, Vincent Conitzer, David Kempe and Detlof von Winterfeldt, not only for serving on my thesis committee but also for providing invaluable suggestions for my research and career. Special thanks to Vincent Conitzer, whom I had the privilege to collaborate with and who has inspired me a lot with his broad knowledge and deep insights. I also thank David Kempe for giving me so much genuine and constructive feedback about my research, writing and presentation in the past five years.

I had the fortune to collaborate with and also learn from many excellent researchers: Bo An, Ashwinkumar Badanidiyuru, Bin Gao, Kshipra Bhawalkar, Matthew Brown, Emma Bowring, Hau Chan, Yu Cheng, Bistra Dilkina, Fei Fang, Benjamin Ford, Rupert Freeman, Jiarui Gan, Mina Guirguis, Manish Jain, Nick Jennings, Albert X. Jiang, Chris Kiekintveld, Kate Larson, Tieyan Liu, Leandro Marcolino, Venil Loyd Noronha, Andrew Plumptre, Zinovi Rabinovich, Eric Rice, Aaron Schlenker, Arunesh Sinha, Solomon Sonya, Omkar Thakoor, Long Tran-Thanh, Phebe Vayanos, Yevgeniy Vorobeychik, Kai Wang, Amulya Yadav and Yue Yin. Particularly, I would like to thank Tieyan Liu and Bin Gao for the memorable time at Microsoft Research Asia and for introducing me to computational game theory. If it were not for them, I do not think

[1] All lists of names in this acknowledgment are in alphabetical order.

I would be pursuing a PhD in computer science. My first publication in this area is under the guidance of Kate Larson; I thank Kate for her patient mentorship. I thank Matthew, Manish and Venil for helping me with the implementation and delivery of the software to the Federal Air Marshal Service (FAMS).

I am grateful to Google Research for the PhD fellowship as well as an enjoyable summer internship, where I had the fortune to learn from many excellent researchers. I also thank Ruggiero Cavallo for hosting me as a summer intern at Yahoo! Labs, which helped to broaden my research scope.

A great advantage of being co-supervised is that I had the chance to interact closely with friends from two large research groups (the Teamcore group and USC theory group). Those I have not mentioned previously include: Yasaman Dehghani Abbasi, Brendan Avent, Joseph Bebel, Biswarup Bhattacharya, Elizabeth Bondi, Hsing-Hau Chen, Ho Yee Cheung, Sarah Cooney, Ehsan Emamjomeh-Zadeh, Shahrzad Gholami, Li Han, Xinran He, Debarun Kar, Lian Liu, Sara Marie Mc Carthy, Thanh Nguyen, Han Ching Ou, Yundi Qian, Ruixin Qiang, Aida Rahmattalabi, Eric Shieh, Alana Shine, Omkar Thakur, Anastasia Voloshinov, Bryan Wilder and Chao Zhang. I truly enjoyed all the casual talks with them, on research, life and other fun topics. All of them together made Teamcore and Theoroom both a home to me.

I am indebted to my parents for their unconditional love, for always respecting and supporting my choices, and for always giving me the best they can give. I am most grateful to Yanqing for always being there with me, sharing my ups and downs. Research time sometimes gets dull and frustrated, but she made every minute of my life joyful.

Contents

List Of Figures

List Of Tables

Abstract

This thesis considers the following question: in systems with self-interested agents (a.k.a., games), how does *information* — i.e., what each agent knows about their environment and other agents' preferences — affect their decision making? The study of the role of information in games has a rich history, and in fact forms the celebrated field of information economics. However, different from previous descriptive study, this thesis takes a *prescriptive* approach and examines computational questions pertaining to the role of information. In particular, it illustrates the double-edged role of information through two threads of research: (1) how to utilize information to one's own advantage in strategic interactions; (2) how to mitigate losses resulting from information leakage to an adversary. In each part, we study the algorithmic foundation of basic models, and also develop efficient solutions to real-world problems arising from physical security domains. Besides pushing the research frontier, the work of this thesis is also directly impacting several real-world applications, resulting in *delivered software* for improving the scheduling of US federal air marshals and the design of patrolling routes for wildlife conservation.

More concretely, the first part of this thesis studies an intrinsic phenomenon in human endeavors termed *persuasion* — i.e., the act of exploiting an informational advantage in order to influence the decisions of others. We start with two real-world motivating examples, illustrating how security agencies can utilize an informational advantage to influence adversaries' decisions and deter potential attacks. Afterwards, we provide a systematic algorithmic study for the foundational economic models underlying these examples. Our analysis not only fully resolves the computational complexity of these models, but also leads to new economic insights. We then leverage the insights and algorithmic ideas from our theoretical analysis to develop new models and solutions for concrete real-world security problems.

The second part of this thesis studies the other side of the double-edged sword, namely, how to deal with disadvantages due to information leakage. We also start with real-world motivating examples to illustrate how classified information about security measures may leak to the adversary and cause significant loss to security agencies. We then propose different models to capture information leakage based on how much the security agency is aware of the leakage situation, and provide thorough algorithmic analysis for these models. Finally, we return to the real-world problems and design computationally efficient algorithms tailored to each security domain.

Part I

Background and Overview

Chapter 1

Overview

1.1 Introduction

This thesis considers a basic question in multi-agent systems: *How does information — i.e., what each agent knows about their environment and other agents' preferences — affect their decision making in systems with self-interested actors (i.e., games)*? Prior work, primarily from the economic literature, reveals that information can have a profound effect on the equilibrium outcome of strategic interactions; in fact, the study of the intricate role of information in games forms the celebrated field of information economics. However, previous study in economics is mostly descriptive while the prescriptive counterpart of these questions have remained largely unexplored. This thesis aims to fill this gap by taking a *prescriptive approach*, and examines *computational questions* pertaining to the role of information in games. In particular, we view information as an endogenous variable of a game and look to design the information structure that induces the most desirable equilibrium. Such problems are intrinsically algorithmic, and are particularly relevant in this digital age given the unprecedented convenience today to generate and communicate information. Our computational study not only results in implementable algorithms that enable automated applications, but also leads to new economic insights regarding the role of information in games.

The primary motivating domain of this thesis is the strategic interaction between a *defender* and an *adversary* in physical security, a.k.a., *security games* (see Figure 1.1 for a few real-world application domains of this thesis). In a security game, the defender must allocate a limited number of security resources, possibly under constraints, to protect a set of targets, while the adversary will strategically choose targets to attack. This important framework has been extensively studied in the past decade, and led to deployed systems in real-world use by security agencies such as the Federal Air Marshal Service (FAMS), the US Coast Guard (USCG) and the Wildlife Conservation Society (WCS) (Tambe, 2011).

(a) Scheduling of federal air marshals

(b) Patrol planning for wildlife conservation

(c) Preventing fare evasion in honor-based metro systems

(d) Flying UAVs to deter poaching in wildlife conservation

Figure 1.1: Concrete security domains which motivate, and are also directly impacted by, the research of this thesis.

Almost all the previous work on security games has focused on optimizing the scheduling of limited security resources to make them unpredictable to a strategic adversary. This thesis, however, departs from previous study by taking a completely different perspective and focuses on studying the effects of information on security games. Obviously, information is playing a more and more important role in security domains today. In fact, at a high level, what the defender does in a security game is essentially to hide information from the adversary by randomization; while the adversary looks to extract information from the defender by conducting surveillance. Additionally, the striking amount of information distilled from today's numerous data sources serves as another key motivating force. For example, defenders and attackers may use sensors, surveillance tools and even infiltration techniques to collect information. Consequently, it is becoming increasingly important for us to understand how information affects such strategic interactions.

1.2 Summary of Contributions

This thesis considers both the positive and negative effects of information on games, illustrating its double-edged role. In particular, we study questions along the following two angles: (1) how to utilize information to one's own advantage in strategic interactions; (2) how to mitigate losses resulting from information leakage to an adversary. Each part begins with real-world motivating examples arising from physical security domains, followed by a systematic study of the fundamental theoretical questions underlying these real-world problems. We then show how our theoretical analyses shed light on practical solutions to the corresponding real-world problem. This forms an organic loop between theory and application.

More concretely, the first part of this thesis studies how an agent can utilize informational advantages in strategic interactions. We start with two motivating examples. The first example illustrates how an unmanned aerial vehicle (UAV) can deter poaching activities by deceptively signaling to poachers the presence of nearby rangers — the information known to the defender but unknown to the poacher. The second example seeks to deter fare evasion in honor-based metro systems[1] via deceptive signaling. Despite the rich literature on security games, such an approach of exploiting informational advantages to improve defense has not received much attention. To study these problems, we start from the intrinsic phenomenon underlying all these examples, which is termed *persuasion*. Specifically, persuasion is the act of exploiting an informational advantage to influence the decisions of others; it has been the theme of a large body of work in economics due to its wide presence in many human activities including security, advertising, marketing, politics, negotiation and financial regulation. We provide a systematic algorithmic study for the most foundational model in this space as well as its natural generalizations, and fully resolve the computational complexity for these models by developing efficient algorithms whose performance match the complexity lower bound. Our algorithmic analysis not only paves the way for applications, but also leads to new economic insights about the problem. Finally, we incorporate these basic models and algorithmic ideas to develop new security game models that capture the aforementioned real-world problems and design practical algorithms to solve the model. En route to these solutions, we overcome additional challenges arising due to particular domain features.

The second part of this thesis considers the other side of the double-edged sword, namely, how to mitigate harms due to information leakage. We also start with two motivating examples. The first example is about the scheduling of US federal air marshals. Since the schedule for each air marshal is usually a round trip, this necessarily creates correlation among the protection statuses of the outbound and return flights. As a result, if the real-time protection status of some

[1] For example, many metro stations in Los Angeles and the Caltrain stations in San Francisco area are honor-based fare collection systems.

outbound flight leaks out to the adversary, he can use this information to infer the protection status of the return flight. This may cause significant loss to the defender if not addressed properly (see Section 7.1 for a concrete example). A similar issue arises in the second motivating example, concerning the design of randomized patrol routes for wildlife conservation. Despite its importance, such vulnerability due to information leakage has not been investigated in the previous work on security games. We initiate the study by developing basic models to capture information leakage and then provide a thorough algorithmic study of the computational complexity for computing the optimal defender strategy. Surprisingly, even for the simplest possible security game model which can be solved by a simple quadratic-time algorithm in the absence of leakage, we show that the problem suddenly becomes computationally intractable when information leakage is considered in the model. This illustrates the intrinsic difficulty in handling leakage. To overcome this complexity barrier, we develop solutions from both theoretical and practical perspectives. On the theoretical side, we design efficient approximation algorithms with provable guarantees; on the practical side, we propose a sampling-based framework which efficiently generates defender mixed strategies with small correlation among targets and thus are robust to leakage. This framework enjoys several practical advantages which make it very useful in the real world. Finally, we use this sampling-framework to develop defender strategies that effectively mitigate the harms arising in the aforementioned real-world problems due to information leakage. En route to these solutions, we overcome specific computational challenges pertaining to each particular domain.

Besides pushing the research frontier, this thesis is also directly impacting several real-world applications. For example, the software based on an algorithm from this thesis for improving the scheduling of US federal air marshal has been delivered to the Federal Air Marsha Service (FAMS) and is currently under pre-deployment evaluation. Another algorithm of this thesis for designing randomized patrol routes has been integrated into PAWS, an anti-poaching software system, and is currently being tested at several national parks in Africa.

1.3 Thesis Structure

The remainder of this thesis is structured as follows. In Chapter 2, we describe the background and preliminaries. Chapter 3 surveys the related work. Afterwards, we move to the first main part of this thesis and study how to utilize informational advantages in strategic interactions. Chapter 4 describes two real-world motiving examples. Chapter 5 provides a systematic algorithmic study for the foundational economic models about the strategic use of information. Chapter 6 returns to the real world, and develops new models and algorithms to improve defense by exploiting the defender's informational advantages at various security settings. Next, we move to the second main part of this thesis and study how to mitigate harms due to information loss to an adversary.

We again start with two real-world motiving examples in Chapter 7. Chapter 8 proposes basic models to capture information leakage, followed with rigorous algorithmic analysis for these models. Chapter 9 returns to real world problems, and seeks to develop efficient and practical algorithms tailored to each concrete security setting. Finally, Chapter 10 concludes the thesis with several open directions.

Chapter 2

Background and Preliminaries

2.1 Information in Games

2.1.1 The Importance of Information in Games

The study of how information affects strategic interactions (i.e., games) dates back to the 1970s. A classic example illustrating the importance of information in games is Akerlof's market for "lemons" (Akerlof, 1970). Akerlof considered the example of the market for used cars where buyers and sellers have asymmetric information regarding car qualities. That is, the seller knows the condition of her own car better than buyers. Akerlof observed that if car buyers cannot distinguish between good cars and bad cars (which are also called "lemons"), and therefore are only willing to pay an average price, this will drive the sellers with high-quality cars out of the market. Knowing this, the buyer will further lower his price, which then drives the sellers of average-quality cars out of the market. At its most extreme, the market would only be left with "lemons". Therefore, the information asymmetry among buyers and sellers has led to an extremely inefficient market. To overcome such inefficiency, one way that has been suggested is to let car sellers "signal" their car quality to buyers so that buyers can distinguish good cars from "lemons". Such signaling could be done, for example, by turning to a trusted third party for quality certification, as most of us do today.

The previous example illustrates that more information may be beneficial to some or all players in a game. The opposite effect is observed in brand advertising. Here, advertisers usually adopt a "semi-transparent" information revelation strategy by highlighting their products' positive attributes while obscuring the defects. In fact, in most economic activities, players or system designers tend to selectively reveal their private information in order to yield a more desirable equilibrium outcome. Such phenomena have been observed and analyzed in numerous economic realms, e.g., advertising (Anderson & Renault, 2006; Waldfogel & Chen, 2006; Johnson & Myatt, 2006; Chakraborty & Harbaugh, 2014), voting (Alonso & Camara, 2014), security (Brown,

Carlyle, Diehl, Kline, & Wood, 2005; Powell, 2007; Zhuang & Bier, 2010), medical research (Kolotilin, 2015), and financial regulation (Gick & Pausch, 2012; Goldstein & Leitner, 2013).

As all these works make clear, *the information structure of a game — i.e., who has what information — can profoundly affect its equilibrium outcome.* This raises a fundamental research question: how should a player utilize her own information advantage to influence the information structure of a game so that a desirable equilibrium outcome is attained? Unsurprisingly, this basic question and its instantiations in concrete domains have been extensively studied in the past. This line of work has led to many models which study how to influence the equilibrium through the control of information in different applications. In the next section, we describe one of the most foundational models in this space, namely the Bayesian persuasion model, which has been a building block of many models and applications including some of the new security games developed in this thesis.

2.1.2 Persuasion by Utilizing Informational Advantages

Persuasion, sometimes also known as *signaling*, is the act of exploiting an informational advantage in order to influence the decisions of others. Persuasive communication is intrinsic in most human activities and has been estimated to account for almost a third of all economic activity in the US (Antioch, 2013). Such scenarios are increasingly common in the information economy. It is therefore unsurprising that persuasion has been the subject of a large body of work in recent years. In the rich literature of persuasion, perhaps no model is more basic and fundamental than the *Bayesian Persuasion* model of (Kamenica & Gentzkow, 2011), generalizing an earlier model from (Brocas & Carrillo, 2007). Here there are two players, who we call the *sender* and the *receiver*. The receiver is faced with selecting one of a number of *actions*, each of which is associated with an a-priori unknown payoff to both players. The *state of nature*, describing the payoff to the sender and receiver from each action, is drawn from a prior distribution known to both players. However, the sender possesses an informational advantage, namely access to the *realized* state of nature prior to the receiver choosing his action. In order to persuade the receiver to take a more favorable action for her, the sender can *commit* to a policy, often known as an *information structure* or *signaling scheme*, of releasing information about the realized state of nature to the receiver before the receiver makes his choice. This policy may be simple, say by always announcing the payoffs of the various actions or always saying nothing, or it may be intricate, involving partial information and added noise. Crucially, the receiver is aware of the sender's committed policy, and moreover is rational and Bayesian.

An Example of Persuasion

To illustrate the intricacy of Bayesian Persuasion, (Kamenica & Gentzkow, 2011) use a simple example in which the sender is a prosecutor, the receiver is a judge, and the state of nature is the guilt or innocence of a defendant. The receiver (judge) has two actions, conviction and acquittal, and wishes to maximize the probability of rendering the correct verdict. On the other hand, the sender (prosecutor) is interested in maximizing the probability of conviction. As they show, it is easy to construct examples in which the optimal signaling scheme for the sender releases noisy partial information regarding the guilt or innocence of the defendant. For example, if the defendant is guilty with probability $\frac{1}{3}$, the prosecutor's best strategy is to claim "guilt" whenever the defendant is guilty, and also claim "guilt" just under half the time when the defendant is innocent. As a result, the defendant will be convicted whenever the prosecutor claims "guilt" (happening with probability just under $\frac{2}{3}$), assuming that the judge is fully aware of the prosecutor's signaling scheme. We note that it is not in the prosecutor's interest to always claim "guilt", since a rational judge aware of such a policy would ascribe no meaning to such a signal, and render his verdict based solely on his prior belief — in this case, this would always lead to acquittal.[1]

2.2 Security Games

2.2.1 The General Security Game Model

The security of critical infrastructures and resources is an important concern around the world, especially given the increasing threats of terrorism. Limited security resources cannot provide full security coverage at all places all the time, leaving potential attackers the chance to explore patrolling patterns and attack the weakness. How can we make use of the limited resources to build the most effective defense against strategic attackers? The past decade has seen an explosion of research in an attempt to address this fundamental question, which has led to the development of the well-known model of security games. The *security game* is a basic model for resource allocation in adversarial environments, and naturally captures the strategic interaction between security agencies and potential adversaries. Indeed, these models and their game-theoretic solutions have led to real-world deployments in use today by major security agencies. For example, they are used by LAX airport for checkpoint placement, the US Coast Guard for port patrolling and the Federal Air Marshal Service for scheduling air marshals (Tambe, 2011). Recently, new models and algorithms have been tested by the Transportation Security Administration for airport

[1] In other words, a signal is an abstract object with no intrinsic meaning, and is only imbued with meaning by virtue of how it is used. In particular, a signal has no meaning beyond the posterior distribution on states of nature it induces.

passenger screening (Brown, Sinha, Schlenker, & Tambe, 2016) and by non-governmental organizations in Malaysia for wildlife protection (Fang, Nguyen, Pickles, Lam, Clements, An, Singh, Tambe, & Lemieux, 2016b).

Next, we give a formal description of security games.

Player Strategies

A security game is a two-player game played between a *defender* and an *attacker*. The defender possesses multiple security resources and aims to allocate these resources to protect n *targets* (e.g., physical facilities, critical locations, etc.) from the attacker's attack. We use $[n]$ to denote the set of these targets. A *defender pure strategy* is a subset of targets that is protected (a.k.a., *covered*) in a feasible allocation of these resources. For example, the defender may have $k(< n)$ security resources, each of which can be assigned to protect any target. In this simple example, any subset of $[n]$ with size at most k is a defender pure strategy. However, in practice, there are usually resource allocation constraints; thus not all such subsets correspond to feasible allocations. We will provide more examples in Section 2.2.3.

A more convenient representation of a pure strategy is a *binary* vector $\mathbf{e} \in \{0,1\}^n$, in which the entries of value 1 specify the covered targets. Let $\mathcal{E} \subseteq \{0,1\}^n$ denote the set of all defender pure strategies. Notice that $\mathcal{E}$ also represents a *set system*. The size of $\mathcal{E}$ is very large, usually exponential in the number of security resources. In the example mentioned above, $|\mathcal{E}| = \Omega(n^k)$ which is exponential in k. Therefore, computational efficiency in security games means time polynomial in n, *not* $|\mathcal{E}|$. A defender mixed strategy is a distribution $\mathbf{p}$ over the elements in $\mathcal{E}$. The attacker chooses one target to attack[2]; thus an *attacker pure strategy* is a target $i \in [n]$. We use $\mathbf{y} \in \Delta_n$ to denote an attacker mixed strategy where y_i is the probability of attacking target i.

Payoff Structures

The payoff structure of the game is as follows: given that the attacker attacks target i, the defender gets utility $U_c^d(i)$ if target i is covered or utility $U_u^d(i)$ if i is uncovered; while the attacker gets utility $U_c^a(i)$ if target i is covered or a reward $U_u^a(i)$ if i is uncovered. Both players have utility 0 on the other $n-1$ unattacked targets. A crucial structure of security games is summarized in the following assumption: $U_c^d(i) > U_u^d(i)$ and $U_c^a(i) < U_u^a(i)$ for all $i \in [n]$. That is, covering a target is strictly beneficial to the defender compared to not covering it; and the attacker prefers to

[2]There are generalizations of security games in which an attacker may attack multiple targets (see, e.g., (Korzhyk, Conitzer, & Parr, 2011a)). However, in all the models considered in this thesis, the attacker attacks only a single target.

attack a target when it is uncovered.[3] The security game is *zero-sum* if $U_c^d(i) + U_c^a(i) = 0$ and $U_u^d(i) + U_u^a(i) = 0$ for all $i \in [n]$.

The defender's utility, as a function of the defender pure strategy $\mathbf{e}$ and attacker pure strategy i, can be formally expressed as

$$U^d(\mathbf{e}, i) = U_c^d(i) \cdot e_i + U_u^d(i) \cdot (1 - e_i),$$

where e_i is the i'th entry of $\mathbf{e}$. Given a defender mixed strategy $\mathbf{p} \in \Delta_{|\mathcal{E}|}$ and attacker mixed strategy $\mathbf{y} \in \Delta_n$, we use $U^d(\mathbf{p}, \mathbf{y})$ to denote the defender's expected utility, which can be expressed as

$$\begin{aligned} U^d(\mathbf{p}, \mathbf{y}) &= \textstyle\sum_{\mathbf{e}\in\mathcal{E}} \sum_{i=1}^n p_e y_i U^d(\mathbf{e}, i) \\ &= \textstyle\sum_{\mathbf{e}\in\mathcal{E}} \sum_{i=1}^n p_e y_i \Big(U_c^d(i) \cdot e_i + U_u^d(i) \cdot (1 - e_i) \Big) \\ &= \textstyle\sum_{i=1}^n y_i \Big(r_i \cdot \sum_{\mathbf{e}\in\mathcal{E}} p_e e_i + c_i \cdot (1 - \sum_{\mathbf{e}\in\mathcal{E}} p_e e_i) \Big) \\ &= \textstyle\sum_{i=1}^n y_i \Big(U_c^d(i) \cdot x_i + U_u^d(i) \cdot [1 - x_i] \Big) \end{aligned} \tag{2.1}$$

where

$$x_i = \sum_{\mathbf{e}\in\mathcal{E}} p_e e_i \in [0, 1] \tag{2.2}$$

is the *marginal* coverage probability of target i. Let $\mathbf{x} = (x_1, \ldots, x_n)^T$ denote the marginal probability for all targets induced by the mixed strategy $\mathbf{p}$. Notice that the marginal probability induced by a pure strategy $\mathbf{e}$ is precisely $\mathbf{e}$ itself. Equation (2.1) shows that the defender's expected utility is *bilinear* in $\mathbf{x}$ and $\mathbf{y}$, where $\mathbf{x}$ is the marginal probability induced by the defender mixed strategy.

The convex hull of $\mathcal{E}$ forms a polytope $\mathcal{P} = \{\mathbf{x} : \mathbf{x} = \sum_{\mathbf{e}\in\mathcal{E}} p_e \cdot \mathbf{e}, \forall \mathbf{p} \in \Delta_{|\mathcal{E}|}\}$ which consists of all the *feasible* (i.e., implementable by a defender mixed strategy) marginal probabilities. Therefore, we will also interpret a point $\mathbf{x} \in \mathcal{P}$ as a mixed strategy, and instead write the defender's utility as $U^d(\mathbf{x}, \mathbf{y})$. Similarly, the attacker's expected utility can be compactly represented in the following form. We note that $U^a(\mathbf{x}, \mathbf{y})$ is also bilinear in $\mathbf{x}$ and $\mathbf{y}$.

$$U^a(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n y_i \Big(U_c^a(i) \cdot [1 - x_i] + U_u^a(i) \cdot x_i \Big).$$

2.2.2 Equilibrium Concepts

Zero-Sum Settings and the Minimax Equilibrium

Many security games, including some deployed systems (An, Shieh, Tambe, Yang, Baldwin, DiRenzo, Maule, & Meyer, 2012; Yin, Jiang, Tambe, Kiekintveld, Leyton-Brown, Sandholm,

[3] In practice, the attacker can also choose to not attack. This can be incorporated into the current model by adding a dummy target. Therefore, we will not explicitly consider this.

& Sullivan, 2012), are modeled as zero-sum games. That is, the defender's reward [cost] is the attacker's cost [reward]. For example, in the deployed security system for patrolling proof-of-payment metro systems (Yin et al., 2012), the defender aims to catch fare evaders at metro stations. This game can be viewed as zero-sum due to the following reasons: the evader's cost of paying a fine is the defender's reward of catching the evader, while the ticket price is the evader's reward and the defender's cost if failing to catch the evader. In zero-sum games, all standard equilibrium concepts are payoff-equivalent to the well-known *minimax equilibrium*.

General-Sum Settings and the Strong Stackelberg Equilibrium

When the game is *not* zero-sum, the main solution concept adopted in the literature of security games is the *strong Stackelberg equilibrium* (**SSE**) (von Stackelberg, 1934; von Stengel & Zamir, 2004). In particular, the defender plays the role of the *leader* and can *commit* to a mixed strategy before the attacker moves. The attacker observes the defender's mixed strategy and best responds. This is motivated by the consideration that the attacker usually does surveillance before committing an attack, and thus is able to observe the empirical distribution of the defender's patrolling strategy (Tambe, 2011). In this case, our goal is to compute the optimal mixed strategy for the defender to commit to (the attacker's best response problem is usually easy). The adoption of SSE has another advantage — in contrast to the intractability of Nash equilibria in normal-form games (Daskalakis, Goldberg, & Papadimitriou, 2006; Chen, Deng, & Teng, 2009), SSE is typically tractable in both normal-form games (Conitzer & Sandholm, 2006) and security games (Xu, 2016). Moreover, previous work shows that under minor technical assumptions, the defender's SSE strategy is always a Nash equilibrium strategy and all Nash equilibria in security games are exchangeable, which alleviates the equilibrium selection issue (Korzhyk, Yin, Kiekintveld, Conitzer, & Tambe, 2011b). This serves as a theoretical justification for adopting SSE in security games. Notice that the classic Stackelberg security game model always assumes that the attacker is not able to observe, even partially, the defender's real-time deployment (i.e., the sampled pure strategy).

2.2.3 Three Concrete Examples

Section 2.2.1 gives an abstract description about the general security game model. The difference among various concrete security games essentially lies in the structure of $\mathcal{E}$, i.e., the structure of pure strategies. Next, we will describe a few examples.

Airport Checkpoint Placement. In the problem of placing checkpoints at different entrances of an airport to prevent potential attackers' attack (Pita, Jain, Marecki, Ordóñez, Portway, Tambe, Western, Paruchuri, & Kraus, 2008a), the defender can place limited security resources at any

subset of airport entrances of a limited size. This can be modeled as a game where the defender has k security resources, and each resource can be assigned to protect any one of n targets. Therefore, any subset of $[n]$ of size at most k is a defender pure strategy. The set system $\mathcal{E}$ is also called a uniform matroid in this setting.

Scheduling of Federal Air Marshals. In the problem of scheduling air marshals to protect flights (Jain, Kardes, Kiekintveld, Ordez, & Tambe, 2010), flights are targets to be protected and air marshals are security resources. Since the schedule for each air marshal is usually a tour consisting of multiple flights, each pure strategy in this case correspond to k *feasible* tours where k is the number of air marshals. Here, tour feasibility means that the departure and arrival time of all flights within the tour should be compatible.

Patrol Route Design for Wildlife Conservation. In the problem of designing randomized patrol routes for wildlife conservation (Fang et al., 2016b), targets correspond to the areas to be patrolled (usually discretized into cells), and rangers are security resources. Any defender pure strategy consists of k feasible patrol routes, each for a ranger.

Chapter 3

Related Work

3.1 Persuasion

To our knowledge, (Brocas & Carrillo, 2007) were the first to explicitly consider persuasion through information control. They consider a sender with the ability to costlessly acquire information regarding the payoffs of the receiver's actions, with the stipulation that acquired information is available to both players. This is technically equivalent to an informed sender who commits to a signaling scheme. Brocas and Carrillo restrict attention to a particular setting with two states of nature and three actions, and characterize optimal policies for the sender and their associated payoffs. The Bayesian Persuasion model of (Kamenica & Gentzkow, 2011) naturally generalizes (Brocas & Carrillo, 2007) to finite (or infinite yet compact) states of nature and action spaces. They establish a number of properties of optimal information structures in this model; most notably, they characterize settings in which signaling strictly benefits the sender in terms of the convexity/concavity of the sender's payoff as a function of the receiver's posterior belief. The Bayesian persuasion model is foundational for understanding the strategic use of informational advantage since it considers essentially the simplest possible scenario in this space — one persuader (i.e., the sender) influences the action of one decision maker (i.e., the receiver).

Since (Brocas & Carrillo, 2007) and (Kamenica & Gentzkow, 2011), an explosion of interest in persuasion problems followed. Generalizations and variants of the Bayesian persuasion model have been considered: (Gentzkow & Kamenica, 2016) consider multiple senders, (Alonso & Câmara, 2016) consider multiple receivers in a voting setting, (Gentzkow & Kamenica, 2014) consider costly information acquisition, (Rayo & Segal, 2010) consider an outside option for the receiver, and (Kolotilin, Mylovanov, Zapechelnyuk, & Li, 2017) considers a receiver with private side information. The basic Bayesian persuasion model underlies, or is closely related to, recent work in a number of different domains: price discrimination (Bergemann, Brooks, & Morris, 2015), advertising (Chakraborty & Harbaugh, 2014), security games (Rabinovich, Jiang, Jain, & Xu, 2015), recommendation systems (Kremer, Mansour, & Perry, 2014; Mansour, Slivkins, &

Syrgkanis, 2015), medical research (Kolotilin, 2015), and financial regulation (Gick & Pausch, 2012; Goldstein & Leitner, 2013).

A important generalization of the Bayesian persuasion model is one recently proposed by (Arieli & Babichenko, 2016). Here the sender interacts with multiple receivers, each of whom is restricted to a binary choice of actions. As mentioned in (Arieli & Babichenko, 2016; Babichenko & Barman, 2017), settings like this arise when a manager seeks to persuade investors to invest in a project, or when a principal persuades opinion leaders in a social network with the goal of maximizing social influence. Each receiver's utility depends only on his own action and the state of nature, but crucially not on the actions of other receivers — the *no externality* assumption. The sender's utility, on the other hand, depends on the state of nature as well as the *profile* of receiver actions. As in (Kamenica & Gentzkow, 2011), the state of nature is drawn from a common prior, and the sender can commit to a policy of revealing information regarding the realization of the state of nature. Since there are multiple receivers, this policy — the information structure — is more intricate, since it can reveal different, and correlated, information to different receivers. As made clear in (Arieli & Babichenko, 2016), such flexibility is crucial to the sender unless receivers are homogeneous and the sender's utility function highly structured (for example, additively separable across receivers). In particular, if restricted to a *public communication channel*, the sender is limited in her ability to discriminate between receivers and correlate their actions, whereas a *private communication channel* provides more flexibility. However, the extent to which a public communication channel limits the sender's powers of persuasion is a fundamental question which has not been thoroughly explored.

Much of the earlier work on persuasion (a.k.a., signaling), in particular its computational aspects, focused on public signaling models. This includes work on signaling in auctions (Emek, Feldman, Gamzu, Paes Leme, & Tennenholtz, 2012; Miltersen & Sheffet, 2012; Guo & Deligkas, 2013; Dughmi, Immorlica, & Roth, 2014), voting (Alonso & Camara, 2014), routing (Bhaskar, Cheng, Ko, & Swamy, 2016), and abstract game models (Dughmi, 2014; Cheng, Cheung, Dughmi, Emamjomeh-Zadeh, Han, & Teng, 2015; Bhaskar et al., 2016; Rubinstein, 2017). The work of (Cheng et al., 2015) is relevant to our results in Section 5.3 in that they identify conditions under which public persuasion problems are tractable to approximate, and prove impossibility results in some cases where those conditions are violated. Our hardness proof in Section 5.3 is in part based on some of their ideas.

Private persuasion has been less thoroughly explored, particularly through the computational lens. There is a recent line of work that explores private persuasion in the context of voting (Wang, 2015; Chan, Gupta, Li, & Wang, 2016; Bardhi & Guo, 2016). Additionally, the space of possible information structures and their induced equilibria is characterized in two-agent two-action games by (Taneva, 2015).

The models in (Kamenica & Gentzkow, 2011; Arieli & Babichenko, 2016) and other works are crucially based on the assumption that the sender has the power of commitment to a signaling scheme. The commitment assumption is not as unrealistic as it might first sound, and a number of arguments to that effect are provided in (Rayo & Segal, 2010; Kamenica & Gentzkow, 2011; Dughmi, 2017). We mention one of those arguments here: commitment arises organically at equilibrium if the sender and receiver(s) interact repeatedly over a long horizon, in which case commitment can be thought of as a proxy for "establishing credibility."

Other Problems Related to Persuasion

Optimal persuasion is a special case of *information structure design* in games. The space of (private channel) information structures is studied by (Bergemann & Morris, 2016), who observe that these information structures and their associated equilibria form a generalization of correlated equilibria, and term the generalization the *Bayes Correlated Equilibrium (BCE)*. Recent work in the CS community has also examined the design of information structures algorithmically. Work by (Emek et al., 2012), (Miltersen & Sheffet, 2012), (Guo & Deligkas, 2013), and (Dughmi et al., 2014), examine optimal signaling in a variety of auction settings, and presents polynomial-time algorithms and hardness results. (Dughmi, 2014) exhibits hardness results for signaling in two-player zero-sum games, and (Cheng et al., 2015) present an algorithmic framework and apply it to a number of different signaling problems.

Also related to the Bayesian persuasion model is the extensive literature on *cheap talk* starting with (Crawford & Sobel, 1982). Cheap talk can be viewed as the analogue of persuasion when the sender cannot commit to an information revelation policy. Crawford and Sobel (1982) characterize the set of Bayesian Nash equilibria of the cheap talk game and show that, under technical assumptions, the sender's equilibrium signaling scheme is more informative when her preference is more aligned with the receiver. After (Crawford & Sobel, 1982), there has been extensive study in the cheap talk model, and its variants and applications; we refer the reader to (Crawford, 1998) for a survey. When the sender has the power to commit, the game becomes a Stackelberg game. The commitment assumption in persuasion has been justified on the grounds that it arises organically in repeated cheap talk interactions with a long horizon — in particular when the sender must balance his short term payoffs with long-term credibility. We refer the reader to the discussion of this phenomenon in (Rayo & Segal, 2010). Also to this point, (Kamenica & Gentzkow, 2011) mention that an earlier model of repeated 2-player games with asymmetric information by (Aumann, Maschler, & Stearns, 1995) is mathematically analogous to Bayesian persuasion.

Various recent models on *selling information* in (Babaioff, Kleinberg, & Paes Leme, 2012; Bergemann & Bonatti, 2015; Bergemann, Bonatti, & Smolin, 2016) are quite similar to Bayesian

persuasion, with the main difference being that the sender's utility function is replaced with revenue. Whereas (Babaioff et al., 2012) consider the algorithmic question of selling information when states of nature are explicitly given as input, the analogous algorithmic questions to ours have not been considered in their model. We speculate that some of our algorithmic techniques might be applicable to models for selling information when the prior distribution on states of nature is represented succinctly.

3.2 Information in Security Games

Secrecy, Deception, and Strategic Signaling in Security

Previous work on homeland security has realized the importance of information asymmetry between the defender and adversary (Brown et al., 2005; Powell, 2007; Zhuang & Bier, 2010). In particular, they justify, via theoretical models, that it is important for the defender to hide private information and remain unpredictable to the adversary. However, these works mainly focused on studying how a defender can hide private information by secrecy and deception. For example, (Powell, 2007) observes that more defense on a particular target may not always be beneficial, since it may help the attacker to infer the importance of a target. These works inspire our exploration of the role of information in Stackelberg security games (SSGs). However, their goals and approaches differ from ours.

(Yin, An, Vorobeychik, & Zhuang, 2013) consider optimal allocation of deceptive resources (e.g., hidden cameras) in the Stackelberg game model. This naturally introduces asymmetric information regarding deployments of resources between the defender and attacker — i.e., the defender has private information regarding the allocation of these security resources while the attacker may not know. However, (Yin et al., 2013) did not consider the strategic use of such informational advantage. Instead, they model the failure of deceptive resources by a probability and feed it to a resource allocation formulation.

(Zhuang & Bier, 2011) study a question that is more relevant to ours. They develop a game-theoretic model to analyze whether a defender should disclose correct information about her resource allocation, incorrect information, or no information. However, their work is more about comparing three natural strategies of information revelation while our work takes an optimization approach to compute the optimal strategy of information revelation.

To the best of our knowledge, little is known in the prior literature about how to optimally reveal a defender's private information to improve the defense. Only very recently, concurrently with this thesis work, have researchers started to investigate the strategic use of signaling or deception to improve the defender's utility in security games (Rabinovich et al., 2015; Talmor & Agmon, 2017; Guo, An, Bosansky, & Kiekintveld, 2017). (Rabinovich et al., 2015) study

how a defender can increase her utility by deceptively revealing her private information about the vulnerability of different targets in order to mislead the attacker. (Rabinovich et al., 2015) analyze the computational complexity of the problem and experimentally show that such strategic use of an informational advantage may significantly increase the defender's utility. (Talmor & Agmon, 2017) compare the advantages and limitations of several different deceptive strategies to manipulate the attacker's belief in a multi-robot adversarial patrolling setting. (Guo et al., 2017) examine the Stackelberg security game setting and analyze the benefit for a defender of disguising her security resources.

Gathering Information via Sensors for Security

The security game model in Section 6.2 uses UAVs (more generally, mobile sensors) to collect information about the poacher and deceptively signal the defender's private information to deter illegal poaching. This part relates to several threads of research on security. The first line of research considers how to use UAVs to gather information or monitor targets (Stranders, De Cote, Rogers, & Jennings, 2013; Mersheeva & Friedrich, 2015). The main research challenge there is to optimize the patrolling path of UAVs so that it maximizes the defender's objective. These works are usually in non-strategic settings and only consider the planning of UAV paths. In contrast, our work falls into a game-theoretic setting with an adversarial attacker. Moreover, we consider the joint task of UAV path planing and deceptive signaling, and seek to compute the globally optimal defending policy.

Another interesting line of research studies adversarial patrolling games with alarm systems (Basilico, De Nittis, & Gatti, 2017b; Basilico, Celli, De Nittis, & Gatti, 2017a), which also utilizes sensors (i.e., alarms) to assist patrollers. The sensors in all these works are *static* (staying at fixed locations) and do not strategically signal. Sensors in our model, however, can strategically signal and are *mobile*. Such mobility gives us the extra flexibility to optimize their (possibly randomized) allocation.

Concerns of Information Leakage in Games

To our knowledge, (Alon, Emek, Feldman, & Tennenholtz, 2013) are the first to study games with information leakage. They focused on two-player zero-sum normal-form games. (Alon et al., 2013) consider a game with two players (player A and B) and assume that not only player A's mixed strategy but also some partial information about her realized pure strategy is known to player B (a situation that is termed *information leakage*). The goal is to compute player A's optimal strategy to play under the leakage model. Even for simple normal-form zero-sum games, they exhibit NP-hardness results for several model variants. The information leakage in our

work has similar meaning to that of (Alon et al., 2013). However, our specific leakage models are directly tied to the particular structure of security games and are different from the abstract leakage models considered in (Alon et al., 2013). Therefore, their hardness results do not directly apply to our settings.

Information leakage has not received much attention in the study of Stackelberg security games. However, in the literature on adversarial patrolling games (APGs), the attacker's real-time surveillance of the defender's pure strategy has been considered (Agmon, Sadov, Kaminka, & Kraus, 2008; Basilico, Gatti, Rossi, Ceppi, & Amigoni, 2009b; Alpern, Morton, & Papadaki, 2011; Bošanský, Lisý, Jakob, & Pěchouček, 2011; Vorobeychik, An, & Tambe, 2012). All these papers study settings of patrols carried out over space and time, i.e., the defender follows a schedule of visits to multiple targets over time. In addition, they assume that an attack action is *not* instantaneous and it takes time for the attacker to execute an attack, during which the defender can interrupt the attacker by visiting the attacked target. Therefore, even if the attacker can fully observe the current position of the defender (in essence, status of *all* targets), he may not have enough time to complete an attack on a target before being interrupted by the defender. The main challenge there is to create patrolling schedules with the smallest possible time between any two target visits. In contrast, our work studies information leakage in Stackelberg security game models, where the attack is *instantaneous* and cannot be interrupted by the defender's resource rescheduling. Furthermore, as may be more realistic in our settings, we assume that information is leaked from a small subset of targets. As a result, our setting necessitates novel models and techniques.

In some settings, a security game with information leakage can be viewed as an extensive-form game (EFG). Though there has been significant progress in solving general-purpose large EFGs recently (Letchford & Conitzer, 2010; Bošanský, Kiekintveld, Lisý, & Pěchouček, 2014; Cermak, Bosansky, Durkota, Lisy, & Kiekintveld, 2016; Cermak, Bošanský, & Lisý, 2017), we did not take this approach because the size of information sets in our game increases exponentially in the number of security resources, time steps and possibly leaking targets. This very quickly makes our problem intractable (see Section 9.1.2 for more details).

Part II

Exploiting Informational Advantages

Chapter 4

Real-World Motivation and Two Illustrative Examples

In this chapter, we will describe two concrete examples motivated from real-world domains that illustrate how informational advantages can be utilized to improve security.

4.1 Motivating Example I: Deterrence of Fare Evasion

Our first example concerns the problem of deterring fare evasion in honor-based metro stations. Such metro systems exist in many cities, e.g., many metro stations in Los Angeles (see Figure 4.1) and the Caltrain stations in San Francisco area are honor-based fare collection systems. One problem of these systems is that some passengers get into the metro without purchasing a ticket. For example, it was estimated that such fare evasion results in a loss of $5.6 million each year in Los Angeles. To prevent such fare evasion, the Los Angeles Sheriff Department (LASD) allocate ticket inspectors to these metro stations. However, the LASD has a very limited number of ticket inspectors and can only inspect a few stations at a time. Naturally, they will allocate the inspectors randomly with the goal of deterring as much fare evasion as possible.

To be concrete, let us consider the following example. The LASD, as the defender, aims to schedule 10 ticket inspectors to protect 50 *identical* (w.r.t. importance) metro stations, namely $t_1, \cdots, t_{50}$. Each ticket inspector can cover one metro station. Therefore, the defender's pure strategies are simply arbitrary subsets of size at most 10 of the 50 stations. For each "potential"

Figure 4.1: An honor-based metro station in Los Angeles.

fare evader, if he indeed does not purchase a ticket, the defender will get utility 2 for catching the evader through inspection and get utility -2 for failing to catch the fare evader. For simplicity, we assume that a fare evader will be caught for sure if the corresponding station is under inspection. Using the security game notations from Section 2.2, this means $U_d^u(t_i) = -2$ and $U_d^c(t_i) = 2$, for all $i = 1, \cdots, 50$. On the other hand, the fare evader gets utility 2 if he is not caught and utility -6 otherwise. That is, $U_a^u(t_i) = 2$ and $U_a^c(t_i) = -6$, for all $i = 1, \cdots, 50$. The fare evader also has the option of choosing to purchase a ticket. In that case, both players get utility 0. We note that, these simple utility numbers are chosen for convenience, and the example is similarly valid when these numbers are substituted by the realistic ones.

We view the problem as a two-player game played between the defender (i.e., the LASD) and a potential fare evader. By symmetry, the optimal defender strategy is to protect each metro station with probability $\frac{10}{50} = 0.2$. This results in an expected attacker utility $0.8 \times 2 + 0.2 \times (-6) = 0.4$, which is greater than 0, the utility of not purchasing a ticket. Therefore, the fare evader will prefer to not purchase a ticket, resulting in defender utility $0.8 \times (-2) + 0.2 \times 2 = -1.2$.

We have computed the Strong Stackelberg Equilibrium (SSE) — traditionally we would be done. However, one interesting question is whether -1.2 is the best possible utility that the defender can achieve. Is there a way to achieve better defender utility? The answer turns out to be "yes". Our approach exploits the asymmetric knowledge of the defensive strategy between the defender and the fare evader — the defender knows more. We show that, surprisingly, the defender can significantly improve her utility by strategically revealing such information.

For any metro station t_i, let X_c [X_u] denote the event that t_i is under inspection [not under inspection]. The defender's mixed strategy results in $\mathbb{P}(X_c) = 0.2$ and $\mathbb{P}(X_u) = 0.8$. Consider a fare evader at some station, w.l.o.g., say station t_1. The fare evader only knows that t_1 is protected with probability 0.2 , while the defender knows precisely whether station t_1 is protected or not. We now design a policy for the defender to strategically reveal part of this information to the fare evader. More concretely, we will sometimes put a warning sign (e.g., a sign like "inspection in progress") at the entrance of the station. Let σ_+ [σ_-] denote the situation that there is a warning sign [no warning sign]. The policy for putting the sign is defined as follows ($\epsilon > 0$ is a negligible positive constant), and we assume that the defender *commits* to this policy:

$$\begin{aligned} \mathbb{P}(\sigma_+|X_c) = 1 \qquad & \mathbb{P}(\sigma_-|X_c) = 0; \\ \mathbb{P}(\sigma_+|X_u) = 3/4 - \epsilon \qquad & \mathbb{P}(\sigma_-|X_u) = 1/4 + \epsilon. \end{aligned}$$

In other words, if t_1 is under inspection, the defender will always announce σ_+; if t_1 is not under inspection, the defender will announce σ_+ with probability $3/4-\epsilon$ and σ_- with probability $1/4+\epsilon$. This is also called a *signaling scheme* and σ_+, σ_- are signals which carry noisy information about the underlying true protection status of the station. We assume that this signaling scheme

is publicly known (thus also known to the fare evader) since passengers may learn it from their past observations.

We analyze the scheme from the fare evader's perspective. If he receives signal σ_+, occurring with probability

$$\mathbb{P}(\sigma_+) = \mathbb{P}(\sigma_+|X_c)\mathbb{P}(X_c) + \mathbb{P}(\sigma_+|X_u)\mathbb{P}(X_u) = 0.8(1-\epsilon),$$

the fare evader infers the following posterior belief via Bayes' rule:

$$\mathbb{P}(X_c|\sigma_+) = \frac{\mathbb{P}(\sigma_+|X_c)\mathbb{P}(X_c)}{\mathbb{P}(\sigma_+)} = \frac{1}{4(1-\epsilon)}$$

and similarly, $\mathbb{P}(X_u|\sigma_+) = \frac{3-4\epsilon}{4(1-\epsilon)}$. Therefore, the fare evader's expected utility for not purchasing a ticket conditioned on σ_+ is

$$\frac{1}{4(1-\epsilon)} \times (-6) + \frac{3-4\epsilon}{4(1-\epsilon)} \times 2 = \frac{-2\epsilon}{1-\epsilon},$$

which is strictly less than 0. Therefore, the fare evader will prefer to purchase a ticket, and both players get utility 0 instead. On the other hand, if the attacker receives signal σ_- (with probability $0.2 + 8\epsilon$), he infers that the station is not under inspection, and thus will not purchase a ticket. In this situation, the defender's utility is -2. As a result, overall the defender receives expected utility $(0.2+8\epsilon) \times (-2) = -0.4 - 16\epsilon$ at target t_1, which is significantly larger than her original utility -1.2 (for a small ϵ). Interestingly, the attacker's expected utility is $(0.2 + 8\epsilon) \times 2 = 0.4 + 16\epsilon$ which essentially equals his SSE utility 0.4 (up to the negligible ϵ).

We remark that the signals σ_+, σ_- have no intrinsic meaning besides the posterior distributions inferred by the fare evader based on the signaling scheme and prior information. Intuitively, by designing signals, the defender identifies a "part" of the prior distribution that is "bad" for both players, i.e., the posterior distribution of σ_+, and signals as much to the fare evader, so that the two players can "cooperate" to avoid it. This is why the defender can do strictly better while the attacker is not worse off.

4.2 Motivating Example II: Combating Poaching

Our second concrete example concerns the protection of conservation areas (Fang, Nguyen, Pickles, Lam, Clements, An, Singh, & Tambe, 2016a). Illegal poaching is a major threat to endangered animals. For example, from 2010 to 2013, within just 2 years, about 20% of animals in Africa were killed due to illegal poaching (Wittemyer, Northrup, Blanc, Douglas-Hamilton, Omondi, & Burnham, 2014). Recently, there has been a rapidly growing trend of using UAVs, or more generally, mobile sensors, to combat poaching (Figure 4.2). Next, we illustrate how a UAV can utilize the defender's informational advantage to deter illegal poaching.

To be concrete, consider the problem where a defender needs to protect 8 conservation areas whose underlying geographic structure is captured by a cycle graph depicted in Figure 4.3 (e.g., they are the border areas of the park): each *vertex* represents an area. Edges indicate the adjacency relation among these areas. The defender has only one patroller. There is a poacher who seeks to attack one area. For simplicity, assume that these 8 areas are of equal importance.

Figure 4.2: Flying UAVs for conservation

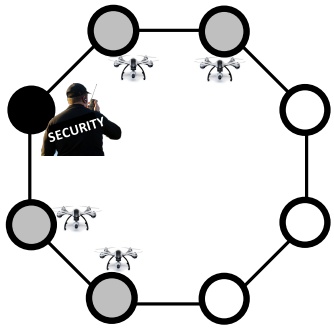


Figure 4.3: Cycle graph.

If the poacher is caught by the patroller in any area, the defender [poacher] gets utility 1 $[-1]$; if the poacher successfully attacks an area, the defender [poacher] gets utility -5 $[1.25]$. The defender has only one patroller, who can protect any area in the graph. Since areas are symmetric, it is easy to see that the optimal patrolling strategy here simply assigns the only patroller to each area with equal probability $1/8$. As a result, the poacher attacks an arbitrary area, resulting in expected defender utility $1 \cdot \frac{1}{8} + (-5) \cdot \frac{7}{8} = -17/4$.

Now consider that the defender is assisted by 4 UAVs (e.g., an NGO named Air Shepherd [http://airshepherd.org/] provides such UAVs for conservation). Each UAV can be assigned to patrol any area. When the poacher visits any area i, he will be caught right away if the patroller is at i. If there is neither the patroller nor a UAV at area i, the poacher will successfully poach animals at that target. If there is a UAV at i, since UAVs are usually visible by the poacher from a distance, the poacher has a chance of choosing to continue the poaching or stop poaching and run away, based on his rational judgment, upon seeing the UAV. If he chooses to continue poaching, the attack will *fail* if the patroller is at any *neighbor* of area i, since the UAV can notify the patroller to come and catch the poacher (e.g., this is how air Shepherd operates). Otherwise, the poaching succeeds (despite the presence of the UAV). The poacher can also choose to stop poaching and immediately run away, in which case both players get utility 0.

We are interested in the defender's optimal strategy for allocating these resources. By symmetry of the problem, it is natural to consider the following randomized strategy. The defender first chooses an area i uniformly at random to place the patroller, and then uses two UAVs to

cover the left two neighbors of i and another two to cover the right two neighbors. The pattern is also illustrated in Figure 4.3 where the thick dark vertex for placing the patroller is chosen uniformly at random. Under such an allocation, each vertex is assigned the patroller with probability $1/8$ and is assigned a UAV with probability $4/8$. By symmetry, the poacher still chooses an arbitrary area to visit. With probability $1/8$, the poacher will be caught by the patroller right away; with probability $3/8$, the poacher encounters neither the patroller nor the UAVs, and thus will successfully conduct an attack. With the remaining $4/8$ probability, the poacher will see a UAV and needs to make a choice of continuing or stopping poaching. Observe that conditioned on a UAV showing up at an area, with probability 0.5, the patroller is at its neighboring area. This is because out of the four areas covered by UAVs, two are neighbors of the patroller-covered area. Therefore, the rational poacher will update his expected utility of continuing poaching, as $(-1) \cdot 0.5 + 1.25 \cdot 0.5 = 0.125$ which is greater than the utility of stopping poaching. So the poacher will prefer to continue poaching, resulting in expected defender utility $1 \cdot 0.5 + (-5) \cdot 0.5 = -2$. Taking expectations over all possible situations, the defender derives expected utility $1 \cdot \frac{1}{8} + (-5) \cdot \frac{3}{8} + (-2) \cdot \frac{4}{8} = -11/4$, which is an improvement over her previous utility of $-17/4$.

A more interesting question is whether the defender can achieve utility that is even larger than $-11/4$. The answer turns out to be "yes". We show that the defender can further improve her utility via *strategic signaling*, which is a natural functionality of UAVs. Such improvement is possible when the poacher visits an area i covered by a UAV. In particular, let θ_{s+} [θ_{s-}] denote the random event that there is a patroller [no patroller] at some neighbor of area i. As mentioned before, conditioned on seeing a UAV at i, the poacher infers $\mathbb{P}(\theta_{s+}) = \mathbb{P}(\theta_{s-}) = 0.5$. However, the UAV will know the precise state of i through communications with the defender. The UAV could strategically signal the state of area i to the poacher with the goal of deterring his poaching. This may sound counter-intuitive at first, but it turns out that strategic signaling does help. In particular, the following signaling scheme with two signals improves the defender's utility:

$$\mathbb{P}(alert|\theta_{s+}) = 1 \qquad \mathbb{P}(quiet|\theta_{s+}) = 0;$$
$$\mathbb{P}(alert|\theta_{s-}) = 0.8 \qquad \mathbb{P}(quiet|\theta_{s-}) = 0.2.$$

That is, when there is a patroller near area i (state θ_{s+}), the UAV always sends an *alert* signal; when there is no patroller near i (state θ_{s-}), 80% percent of the time the UAV still sends an *alert* signal while it keeps *quiet* otherwise.

We assume that the poacher is aware of the signaling scheme and will best respond to each signal. If he receives an *alert* signal, which occurs with probability: $\mathbb{P}(alert) = \mathbb{P}(alert|\theta_{s+})\mathbb{P}(\theta_{s+}) + \mathbb{P}(alert|\theta_{s-})\mathbb{P}(\theta_{s-}) = 0.9$, the poacher infers a posterior distribution on the state by Bayes rule: $\mathbb{P}(\theta_{s+}|alert) = \frac{\mathbb{P}(alert|\theta_{s+})\mathbb{P}(\theta_{s+})}{\mathbb{P}(alert)} = \frac{5}{9}$ and $\mathbb{P}(\theta_{s-}|alert) = \frac{4}{9}$. This

posterior results in expected poacher utility $\frac{5}{9} \cdot (-1) + \frac{4}{9} \cdot 1.25 = 0$, which is the same as the utility from not attacking. We assume that the poacher breaks ties in favor of the defender (see justifications later) and, in this case, chooses to stop poaching. This results in utility 0 for both players. On the other hand, if the poacher receives a *quiet* signal, he knows for sure that there is no patroller nearby; thus he chooses to continue poaching, resulting in defender utility -5. As a result, the above signaling scheme (which occurs whenever a poacher encounters a UAV) results in defender utility $0 \cdot 0.9 + (-5) \cdot 0.1 = -0.5$. Overall, the defender's expected utility is further improved to $1 \cdot \frac{1}{8} + (-5) \cdot \frac{3}{8} + (-0.5) \cdot \frac{4}{8} = -2$, which is less than half of the original loss $-17/4$.

Remark. This example shows how a defender can utilize an informational advantage to deceive the poacher and improve her utility. Note that a signal takes effect only through its underlying posterior distribution over Θ_s. In the above example, the poaching would not have been deterred if the UAV *always* sent an *alert* signal since in that case the poacher would ignore the signal and act based on his prior belief. However, the signals could be *deceptive* in the sense that an *alert* may be issued even when there is no patroller nearby. The poacher still prefers to stop poaching even though he is aware of the potential deception!

Chapter 5

Persuasion and Its Algorithmic Foundation

Though the two motivating examples in Chapter 4 are in security domains, the underlying phenomenon they illustrate is more general and fundamental. Such act of exploiting an informational advantage in order to influence the decisions of others is called *persuasion*. Indeed, persuasion is intrinsic in most human activities — persuasive communication has been estimated to account for almost a third of all economic activity in the US (Antioch, 2013). Such scenarios are increasingly common in today's information economy. It is therefore unsurprising that persuasion has been the subject of a large body of work in recent years. In the rich literature of persuasion, perhaps no model is more basic than the *Bayesian Persuasion* (BP) model of (Kamenica & Gentzkow, 2011). It has been a building block of many models and applications.

In the next section, we will provide a formal description of the BP mode, referring back to the poaching example in Section 4.2 to illustrate how the interaction there can be framed using the BP model. The correspondence between the fare evasion example in Section 4.1 and the BP model follows similarly.

5.1 The Bayesian Persuasion Model

In a Bayesian persuasion game, there are two players: a *sender* and a *receiver*. The receiver is faced with selecting an action from $[n] = \{1, \ldots, n\}$, with an a-priori-unknown payoff to each of the sender and receiver. We assume that payoffs are a function of an unknown *state of nature* θ, drawn from an abstract set Θ of potential realizations of nature. Specifically, the sender and receiver's payoffs are functions $s, r : \Theta \times [n] \to \mathbb{R}$, respectively. We use $\boldsymbol{r} = \boldsymbol{r}(\theta) \in \mathbb{R}^n$ to denote the receiver's payoff vector as a function of the state of nature, where $r_i(\theta)$ is the receiver's payoff if he takes action i and the state of nature is θ. Similarly $\boldsymbol{s} = \boldsymbol{s}(\theta) \in \mathbb{R}^n$ denotes the sender's payoff vector, and $s_i(\theta)$ is the sender's payoff if the receiver takes action i and the state is θ. Without loss of generality, we often conflate the abstract set Θ indexing states of nature with the set of realizable payoff vector pairs $(\boldsymbol{s}, \boldsymbol{r})$ — i.e., we think of Θ as a subset of $\mathbb{R}^n \times \mathbb{R}^n$.

Correspondence to the example of Section 4.2: *In the example, the defender is the sender and the poacher is the receiver. After seeing the UAV, the poacher has two actions — either choose to continue poaching or stop poaching and run away. The random state of nature θ describes whether a ranger is nearby or not, so θ has two possible realizations. Naturally, θ affects both the defender's and poacher's utilities.*

In Bayesian persuasion, it is assumed that the state of nature is a-priori unknown to the receiver, and drawn from a common-knowledge prior distribution λ supported on Θ. The sender, on the other hand, has access to the realization of θ, and can commit to a policy of partially revealing information regarding its realization before the receiver selects his action. Specifically, the sender commits to a *signaling scheme* φ, mapping (possibly randomly) states of nature Θ to a family of *signals* Σ. For $\theta \in \Theta$, we use $\varphi(\theta)$ to denote the (possibly random) signal selected when the state of nature is θ. Moreover, we use $\varphi(\theta, \sigma)$ to denote the probability of selecting the signal σ given a state of nature θ. An algorithm *implements* a signaling scheme φ if it takes as input a state of nature θ, and samples the random variable $\varphi(\theta)$.

Correspondence to the example of Section 4.2: *In the example, the state of nature θ, i.e., whether a ranger is nearby or not, is known to the defender but unknown to the poacher. However, the probability that a ranger is nearby is publicly known. The defender commits to a signaling scheme to deceptively send the warning signal. The process can be randomized since when the ranger is not nearby, the defender sometimes sends the warning signal and sometimes does not.*

Given a signaling scheme φ with signals Σ, each signal $\sigma \in \Sigma$ is realized with probability $\alpha_\sigma = \sum_{\theta \in \Theta} \lambda_\theta \varphi(\theta, \sigma)$. Conditioned on the signal σ, the expected payoffs to the receiver of the various actions are summarized by the vector $\boldsymbol{r}(\sigma) = \frac{1}{\alpha_\sigma} \sum_{\theta \in \Theta} \lambda_\theta \varphi(\theta, \sigma) \boldsymbol{r}(\theta)$. Similarly, the sender's payoffs as a function of the receiver's action are summarized by $\boldsymbol{s}(\sigma) = \frac{1}{\alpha_\sigma} \sum_{\theta \in \Theta} \lambda_\theta \varphi(\theta, \sigma) \boldsymbol{s}(\theta)$. On receiving a signal σ, the receiver performs a Bayesian update and selects an action $i^*(\sigma) \in \operatorname{argmax}_i r_i(\sigma)$ with expected receiver utility $\max_i r_i(\sigma)$. This induces utility $s_{i^*(\sigma)}(\sigma)$ for the sender. In the event of ties when selecting $i^*(\sigma)$, we assume those ties are broken in favor of the sender.

Correspondence to the example of Section 4.2: *When the poacher receives a warning signal, he updates his belief about the probability of a ranger nearby and then best responds.*

We will adopt the perspective of a sender looking to design φ to maximize her expected utility $\sum_\sigma \alpha_\sigma s_{i^*(\sigma)}(\sigma)$, in which case we say φ is *optimal*. When φ yields expected sender utility within an additive [multiplicative] ϵ of the best possible, we say it is *ϵ-optimal* [ϵ-approximate] in the additive [multiplicative] sense. A simple revelation-principle style argument (Kamenica & Gentzkow, 2011) shows that an optimal signaling scheme need not use more than n signals, with one *recommending* each action. Such a *direct* scheme φ has signals $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$, and

satisfies $r_i(\sigma_i) \geq r_j(\sigma_i)$ for all $i, j \in [n]$. We think of σ_i as a signal recommending action i, and the requirement $r_i(\sigma_i) \geq \max_j r_j(\sigma_i)$ as an *persuasiveness* constraint on the signaling scheme — i.e., the recommended action is indeed the receiver's favorite action.[1] All the signaling schemes considered in this thesis will be direct, unless explicitly stated otherwise.[2]

Correspondence to the example of Section 4.2: *The optimal scheme we described in the example uses two signals θ_+, θ_- since the poacher has only two actions. Moreover, θ_+ [θ_-] can be thought of as a persuasive recommendation of stopping [continuing] poaching. So, the scheme we describe is direct.*

Next we mention a few remarks about the results in the next sections. For our results in Section 5.2.4, we relax the persuasiveness constraints by assuming that the receiver follows the recommendation so long as it approximately maximizes his utility — for a parameter $\epsilon > 0$, we relax our requirement to $r_i(\sigma_i) \geq \max_j r_j(\sigma_i) - \epsilon$, which translates to the relaxed persuasiveness constraints $\sum_{\theta \in \Theta} \lambda_\theta \varphi(\theta, \sigma_i) r_i(\theta) \geq \sum_{\theta \in \Theta} \lambda_\theta \varphi(\theta, \sigma_i)(r_j(\theta) - \epsilon)$ in LP (5.1). We call such schemes *ϵ-persuasive*. We judge the suboptimality of an ϵ-persuasive scheme relative to the best (absolutely) persuasive scheme; i.e., in a bi-criteria sense.

We note that expected utilities, persuasiveness, and optimality are properties not only of a signaling scheme φ, but also of the distribution λ over its inputs. Therefore, we often say that a signaling scheme φ is persuasive [ϵ-persuasive] for λ, or optimal [ϵ-optimal] for λ. We also use $u_s(\varphi, \lambda)$ to denote the expected sender utility $\sum_{\theta \in \Theta} \sum_{i=1}^{n} \lambda_\theta \varphi(\theta, \sigma_i) s_i(\theta)$.

The Commitment Assumption.

We conclude this section with a few justifications about the commitment assumption in the persuasion model. The commitment to signaling schemes is justified on the grounds of repeated games with a long horizon — in particular when the sender must balance his short-term payoffs with long-term credibility. We refer the reader to the discussion of this phenomenon in (Rayo & Segal, 2010). Also, (Kamenica & Gentzkow, 2011) mention that an earlier model of repeated 2-player games with asymmetric information by (Aumann et al., 1995) is mathematically analogous to Bayesian persuasion.

With respect to the concrete security applications we study, the commitment to signaling schemes is usually natural and realistic. For example, in the poaching example of Section 4.2, the signaling schemes need to be implemented as software in the UAVs. Once the code is finalized and deployed, the defender is committed to using the signaling scheme prescribed by the code. We will also assume that the receiver (i.e., attacker in security games) is aware of the signaling

[1] Persuasiveness has also been called *incentive compatibility* or *obedience* in prior work.

[2] One reason is that schemes for which it is tractable to compute the best receiver response (or the desired ϵ-best response) are w.l.o.g direct. Therefore, indirect schemes are somewhat less useful to consider.

scheme and will best respond to each signal. This is because by interacting with the system, the attacker can gradually learn each signal's posterior. This is particularly true in "green security" domains which generally involve limited penalties for being caught (Carthy, Tambe, Kiekintveld, Gore, & Killion, 2016; Fang et al., 2016a). Moreover, there is usually a community of attackers who can learn these probabilities by sharing knowledge.

5.2 Algorithmic Foundation for Bayesian Persuasion

Naturally, the sender in the Bayesian persuasion model seeks to find the signaling scheme that maximizes her expected utility subject to the receiver's strategic response. Therefore, the Bayesian persuasion problem is an optimization problem by nature. We now provide a thorough algorithmic analysis for the model and pin down the complexity of the problem under several natural input models. To the best of our knowledge, this is the first algorithmic study for this foundational economic model. Our results not only pave the way to applications and help to operationalize the model, but also provide structural insights into the problem. Moreover, complexity-theoretic results often shed light on whether or not a model is realistic.

5.2.1 Explicit Input Model

We start with the simple case where the distribution for the state of nature θ is explicitly given. That is, the probability for each state of nature is explicitly enumerated. In this case the sender's optimization problem can be formulated as a linear program (LP) with variables $\{\varphi(\theta, \sigma_i) : \theta \in \Theta, i \in [n]\}$.

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta}\sum_{i=1}^{n} \lambda_\theta \varphi(\theta,\sigma_i) s_i(\theta) & \\ \text{subject to} & \sum_{\theta\in\Theta} \lambda_\theta \varphi(\theta,\sigma_i) r_i(\theta) \geq \sum_{\theta\in\Theta} \lambda_\theta \varphi(\theta,\sigma_i) r_j(\theta), & \text{for } i,j \in [n]. \\ & \sum_{i=1}^{n} \varphi(\theta,\sigma_i) = 1, & \text{for } \theta \in \Theta. \\ & \varphi(\theta,\sigma_i) \geq 0, & \text{for } \theta \in \Theta, i \in [n]. \end{array} \tag{5.1}$$

At a high level, the LP maximizes the sender's expected utility subject to that the recommendation of each signal is persuasive and the scheme is feasible. This shows that the optimal persuasion problem can be solved in polynomial time for the explicit input model since linear programs can be solved in polynomial time in the LP size.

5.2.2 Poly-Time Solvability for Persuasion with I.I.D Actions

In this section, we assume that the payoffs of different actions are independently and identically distributed (i.i.d.) according to an explicitly-described marginal distribution. To better motivate this setting, we start with an illustrative example.

Example 1 (An Example of Persuasion with I.I.D. Actions)**.** *Our example is in the context of wildlife conservation. The receiver is a poacher, actions correspond to visiting conservation areas for poaching, and the sender is a security agency or defender with access to statuses of conservation areas (e.g., animal populations, ranger locations, etc.) which are a priori unknown to the poacher. The misaligned incentives between the defender and poacher give rise to a nontrivial Bayesian persuasion problem. In fact, interesting examples exist when statuses of conservation areas are independent from each other, or even i.i.d. Consider the following simple example which fits into the i.i.d. model considered in this section: there are two conservation areas, each of which is a priori equally likely to be in one of the following three states (independently): protected animals show up and rangers are patrolling the area (state A); protected animals show up and rangers are not patrolling the area (state B); only regular animals — which are not protected — show up (state C). We refer to A/B/C as the* types *of an area, and associate them with poacher utilities of* -1, 1, *and* ϵ, *respectively. Suppose that the defender's goal is to prevent the poacher to attack an area with protected animals. Concretely, the defender receives utility* -1 *if the poacher attacks an area of type A or B,*[3] *and utility* 0 *if the poacher attacks an area of type C. The poacher will always choose one of these two areas to attack. A simple calculation shows that providing full information to the poacher results in an expected defender utility of* $-\frac{2}{3}$*, as does providing no information. An optimal signaling scheme, which guarantees that the poacher attacks an area with type C whenever such an area exists, is the following: when exactly one of the areas has type C "recommend" that area to the poacher, and otherwise "recommend" any area uniformly at random. A simple calculation using Bayes' rule shows that the poacher prefers to follow the recommendations of this partially informative scheme, and it follows that the expected defender utility is* $-\frac{4}{9}$.

More formally, in the Bayesian persuasion model with i.i.d. actions, each state of nature θ is a vector in $\Theta = [m]^n$ for a parameter m, where $\theta_i \in [m]$ is the *type* of action i. Associated with each type $j \in [m]$ is a pair $(\xi_j, \rho_j) \in \mathbb{R}^2$, where ξ_j [ρ_j] is the payoff to the sender [receiver] when the receiver chooses an action with type j. We are given a marginal distribution over types, described by a vector $\boldsymbol{q} = (q_1, \ldots, q_m) \in \Delta_m$. We assume each action's type is drawn independently according to $\boldsymbol{q}$; specifically, the prior distribution λ on states of nature is given by $\lambda(\theta) = \prod_{i \in [n]} q_{\theta_i}$. For convenience, we let $\boldsymbol{\xi} = (\xi_1, \ldots, \xi_m) \in \mathbb{R}^m$ and $\boldsymbol{\rho} = (\rho_1, \ldots, \rho_m) \in \mathbb{R}^m$ denote the type-indexed vectors of sender and receiver payoffs, respectively. We assume $\boldsymbol{\xi}$, $\boldsymbol{\rho}$, and $\boldsymbol{q}$ — the parameters describing an i.i.d. persuasion instance — are given explicitly.

[3] *The defender does not want the poacher to attack an area with protected animals even though there are patrollers there (i.e., in state B). This is because the protected animals may have already be killed before the rangers catch the poacher, and this is a huge loss to the defender.*

$$
\begin{array}{ll}
M^{\sigma_i} = \sum_\theta \lambda(\theta)\varphi(\theta,\sigma_i)M^\theta, & \text{for } i = 1,\dots,n. \\
\sum_{i=1}^n \varphi(\theta,\sigma_i) = 1, & \text{for } \theta \in \Theta. \\
\varphi(\theta,\sigma_i) \geq 0, & \text{for } \theta \in \Theta, i \in [n].
\end{array}
$$

Figure 5.1: Realizable signatures $\mathcal{P}$

$$
\begin{array}{lll}
\max & \sum_{i=1}^n \boldsymbol{\xi} \cdot M_i^{\sigma_i} & \\
\text{s.t.} & \boldsymbol{\rho} \cdot M_i^{\sigma_i} \geq \boldsymbol{\rho} \cdot M_j^{\sigma_i}, & \text{for } i,j \in [n]. \\
& (M^{\sigma_1},\dots,M^{\sigma_n}) \in \mathcal{P} &
\end{array}
$$

Figure 5.2: Persuasion in signature space

Note that the number of states of nature is m^n, and therefore the natural representation of a signaling scheme has nm^n variables. As a result, the natural linear program for the persuasion problem in Section 5.2.1 has an exponential in n number of both variables and constraints. Nevertheless, we will not need to explicitly write down the signaling scheme. Instead, as mentioned in Section 5.1, we seek only to implement an optimal or near-optimal scheme φ as an oracle which takes as input θ and samples a signal $\sigma \sim \varphi(\theta)$. Our algorithms will run in time polynomial in n and m, and will optimize over a space of succinct "reduced forms" for signaling schemes which we term *signatures*, to be described next.

For a state of nature θ, define the matrix $M^\theta \in \{0,1\}^{n\times m}$ so that $M^\theta_{ij} = 1$ if and only if action i has type j in θ (i.e. $\theta_i = j$). Given an i.i.d prior λ and a signaling scheme φ with signals $\Sigma = \{\sigma_1,\dots,\sigma_n\}$, for each $i \in [n]$ let $\alpha_i = \sum_\theta \lambda(\theta)\varphi(\theta,\sigma_i)$ denote the probability of sending σ_i, and let $M^{\sigma_i} = \sum_\theta \lambda(\theta)\varphi(\theta,\sigma_i)M^\theta$. Note that $M^{\sigma_i}_{jk}$ is the joint probability that action j has type k and the scheme outputs σ_i. Also note that each row of M^{σ_i} sums to α_i, and the jth row represents the un-normalized posterior type distribution of action j given signal σ_i. We call $\mathcal{M} = (M^{\sigma_1},\dots,M^{\sigma_n}) \in \mathbb{R}^{n\times m\times n}$ the *signature* of φ. The sender's objective and receiver's persuasiveness constraints can both be expressed in terms of the signature. In particular, using M_j to denote the jth row of a matrix M, the persuasiveness constraints are $\boldsymbol{\rho} \cdot M_i^{\sigma_i} \geq \boldsymbol{\rho} \cdot M_j^{\sigma_i}$ for all $i,j \in [n]$, and the sender's expected utility assuming the receiver follows the scheme's recommendations is $\sum_{i\in[n]} \boldsymbol{\xi} \cdot M_i^{\sigma_i}$.

We say $\mathcal{M} = (M^{\sigma_1},\dots,M^{\sigma_n}) \in \mathbb{R}^{n\times m\times n}$ is *realizable* if there exists a signaling scheme φ with $\mathcal{M}$ as its signature. Realizable signatures constitutes a polytope $\mathcal{P} \subseteq \mathbb{R}^{n\times m\times n}$, which has an exponential-sized extended formulation as shown Figure 5.1. Given this characterization, the sender's optimization problem can be written as a linear program in the space of signatures, shown in Figure 5.2:

Symmetry of the Optimal Signaling Scheme

We now show that there always exists a "symmetric" optimal scheme when actions are i.i.d. Given a signature $\mathcal{M} = (M^{\sigma_1}, \ldots, M^{\sigma_n})$, it will sometimes be convenient to think of it as the set of pairs $\{(M^{\sigma_i}, \sigma_i)\}_{i\in[n]}$.

Definition 1. *A signaling scheme φ with signature $\{(M^{\sigma_i}, \sigma_i)\}_{i\in[n]}$ is* symmetric *if there exist $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$ such that $M_i^{\sigma_i} = \boldsymbol{x}$ for all $i \in [n]$ and $M_j^{\sigma_i} = \boldsymbol{y}$ for all $j \neq i$. The pair $(\boldsymbol{x}, \boldsymbol{y})$ is the* s-signature *of φ.*

In other words, a symmetric signaling scheme sends each signal with equal probability $||\boldsymbol{x}||_1$, and induces only two different posterior type distributions for actions: $\frac{\boldsymbol{x}}{||\boldsymbol{x}||_1}$ for the recommended action, and $\frac{\boldsymbol{y}}{||\boldsymbol{y}||_1}$ for the others. We call $(\boldsymbol{x}, \boldsymbol{y})$ *realizable* if there exists a signaling scheme with $(\boldsymbol{x}, \boldsymbol{y})$ as its s-signature. The family of realizable s-signatures constitutes a polytope $\mathcal{P}_s$, and has an extended formulation by adding the variables $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$ and constraints $M_i^{\sigma_i} = \boldsymbol{x}$ and $M_j^{\sigma_i} = \boldsymbol{y}$ for all $i, j \in [n]$ with $i \neq j$ to the extended formulation of (asymmetric) realizable signatures from Figure 5.1.

We make two simple observations regarding realizable s-signatures. First, $||\boldsymbol{x}||_1 = ||\boldsymbol{y}||_1 = \frac{1}{n}$ for each $(\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{P}_s$, and this is because both $||\boldsymbol{x}||_1$ and $||\boldsymbol{y}||_1$ equal the probability of each of the n signals. Second, since the signature must be consistent with prior marginal distribution $\boldsymbol{q}$, we have $\boldsymbol{x} + (n-1)\boldsymbol{y} = \sum_{i=1}^n M_1^{\sigma_i} = \boldsymbol{q}$. We show that the restriction to symmetric signaling schemes will not reduce the sender's optimal utility.

Theorem 5.2.1. *When the action payoffs are i.i.d., there exists an optimal and persuasive signaling scheme which is symmetric.*

Theorem 5.2.1 is proved in Appendix A.1.1. At a high level, we show that optimal signaling schemes are closed with respect to two operations: *convex combination* and *permutation*. Specifically, a convex combination of realizable signatures — viewed as vectors in $\mathbb{R}^{n\times m\times n}$ — is realized by the corresponding "random mixture" of signaling schemes, and this operation preserves optimality. The proof of this fact follows easily from the fact that linear program in Figure 5.2 has a convex family of optimal solutions. Moreover, given a permutation $\pi \in SS_n$ and an optimal signature $\mathcal{M} = \{(M^{\sigma_i}, \sigma_i)\}_{i\in[n]}$ realized by signaling scheme φ, the "permuted" signature $\pi(\mathcal{M}) = \{(\pi M^{\sigma_i}, \sigma_{\pi(i)})\}_{i\in[n]}$ — where premultiplication of a matrix by π denotes permuting the rows of the matrix — is realized by the "permuted" scheme $\varphi_\pi(\theta) = \pi(\varphi(\pi^{-1}(\theta)))$, which is also optimal. The proof of this fact follows from the "symmetry" of the (i.i.d.) prior distribution about the different actions. Theorem 5.2.1 is then proved constructively as follows: given a realizable optimal signature $\mathcal{M}$, the "symmetrized" signature $\overline{\mathcal{M}} = \frac{1}{n!}\sum_{\pi\in SS_n} \pi(\mathcal{M})$ is realizable, optimal, and symmetric.

Implementing the Optimal Signaling Scheme

We now exhibit a polynomial-time algorithm for persuasion in the i.i.d. model. Theorem 5.2.1 permits re-writing the optimization problem in Figure 5.2 as follows, with variables $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^m$.

$$\begin{aligned} \text{maximize} \quad & n\boldsymbol{\xi} \cdot \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{\rho} \cdot \boldsymbol{x} \geq \boldsymbol{\rho} \cdot \boldsymbol{y} \\ & (\boldsymbol{x}, \boldsymbol{y}) \in \mathcal{P}_s \end{aligned} \tag{5.2}$$

Problem (5.2) cannot be solved directly, since $\mathcal{P}_s$ is defined by an extended formulation with exponentially many variables and constraints, as described previously. Nevertheless, we make use of a connection between symmetric signaling schemes and single-item auctions with i.i.d. bidders to solve (5.2) using the Ellipsoid method. Specifically, we show a one-to-one correspondence between symmetric signatures and (a subset of) symmetric reduced forms of single-item auctions with i.i.d. bidders, defined as follows.

Definition 2. *(Border, 1991) Consider a single-item auction setting with n i.i.d. bidders and m types for each bidder, where each bidder's type is distributed according to $\boldsymbol{q} \in \Delta_m$. An* allocation rule *is a randomized function A mapping a type profile $\theta \in [m]^n$ to a winner $A(\theta) \in [n] \cup \{*\}$, where $*$ denotes not allocating the item. We say the allocation rule has* symmetric reduced form *$\boldsymbol{\tau} \in [0,1]^m$ if for each bidder $i \in [n]$ and type $j \in [m]$, τ_j is the conditional probability of i receiving the item given that she has type j.*

When $\boldsymbol{q}$ is clear from context, we say $\boldsymbol{\tau}$ is *realizable* if there exists an allocation rule with $\boldsymbol{\tau}$ as its symmetric reduced form. We say an algorithm *implements* an allocation rule A if it takes as input θ, and samples $A(\theta)$.

Theorem 5.2.2. *Consider the Bayesian Persuasion problem with n i.i.d. actions and m types, with parameters $\boldsymbol{q} \in \Delta_m$, $\boldsymbol{\xi} \in \mathbb{R}^m$, and $\boldsymbol{\rho} \in \mathbb{R}^m$ given explicitly. An optimal and persuasive signaling scheme can be implemented in $\mathrm{poly}(m, n)$ time.*

Theorem 5.2.2 is a consequence of the following set of lemmas.

Lemma 1. *Let $(\boldsymbol{x}, \boldsymbol{y}) \in [0,1]^m \times [0,1]^m$, and define $\boldsymbol{\tau} = (\frac{x_1}{q_1}, \ldots, \frac{x_m}{q_m})$. The pair $(\boldsymbol{x}, \boldsymbol{y})$ is a realizable s-signature if and only if (a) $||\boldsymbol{x}||_1 = \frac{1}{n}$, (b) $\boldsymbol{x} + (n-1)\boldsymbol{y} = \boldsymbol{q}$, and (c) $\boldsymbol{\tau}$ is a realizable symmetric reduced form of an allocation rule with n i.i.d. bidders, m types, and type distribution $\boldsymbol{q}$. Moreover, assuming $\boldsymbol{x}$ and $\boldsymbol{y}$ satisfy (a), (b) and (c), and given black-box access to an allocation rule A with symmetric reduced form $\boldsymbol{\tau}$, a signaling scheme with s-signature $(\boldsymbol{x}, \boldsymbol{y})$ can be implemented in $\mathrm{poly}(n, m)$ time.*

Lemma 2. *An optimal realizable s-signature, as described by LP* (5.2)*, is computable in $\mathrm{poly}(n, m)$ time.*

Lemma 3. *(See (Cai, Daskalakis, & Weinberg, 2012; Alaei, Fu, Haghpanah, Hartline, & Malekian, 2012)) Consider a single-item auction setting with n i.i.d. bidders and m types for each bidder, where each bidder's type is distributed according to $\boldsymbol{q} \in \Delta_m$. Given a realizable symmetric reduced form $\boldsymbol{\tau} \in [0,1]^m$, an allocation rule with reduced form $\boldsymbol{\tau}$ can be implemented in $\mathrm{poly}(n,m)$ time.*

The proofs of Lemmas 1 and 2 can be found in Appendix A.1.2. The proof of Lemma 1 builds a correspondence between s-signatures of signaling schemes and certain reduced-form allocation rules. Specifically, actions correspond to bidders, action types correspond to bidder types, and signaling σ_i corresponds to assigning the item to bidder i. The expression of the reduced form in terms of the s-signature then follows from Bayes' rule. Lemma 2 follows from Lemma 1, the ellipsoid method, and the fact that symmetric reduced forms admit an efficient separation oracle (see (Border, 1991, 2007; Cai et al., 2012; Alaei et al., 2012)).

A "Simple" $(1 - \frac{1}{e})$-Approximate Scheme

Our next result is a "simple" signaling scheme which obtains a $(1 - 1/e)$ multiplicative approximation when payoffs are nonnegative. This algorithm has the distinctive property that it signals *independently* for each action, and therefore implies that approximately optimal persuasion can be parallelized among multiple colluding senders, each of whom only has access to the type of one or more of the actions.

Recall that an s-signature $(\boldsymbol{x}, \boldsymbol{y})$ satisfies $||\boldsymbol{x}||_1 = ||\boldsymbol{y}||_1 = \frac{1}{n}$ and $\boldsymbol{x} + (n-1)\boldsymbol{y} = \boldsymbol{q}$. Our simple scheme, shown in Algorithm 1, works with the following explicit linear programming relaxation of optimization problem (5.2).

$$\begin{aligned} \text{maximize} \quad & n\boldsymbol{\xi} \cdot \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{\rho} \cdot \boldsymbol{x} \geq \boldsymbol{\rho} \cdot \boldsymbol{y} \\ & \boldsymbol{x} + (n-1)\boldsymbol{y} = \boldsymbol{q} \\ & ||\boldsymbol{x}||_1 = \tfrac{1}{n} \\ & \boldsymbol{x}, \boldsymbol{y} \geq 0 \end{aligned} \tag{5.3}$$

Algorithm 1 has a simple and instructive interpretation. It computes the optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ to the relaxed problem (5.3), and uses this solution as a guide for signaling *independently* for each action. The algorithm selects, independently for each action i, a component signal $o_i \in \{\mathbf{HIGH}, \mathbf{LOW}\}$. Each o_i is chosen so that $\mathbf{Pr}[o_i = \mathbf{HIGH}] = \frac{1}{n}$, and moreover the events $o_i = \mathbf{HIGH}$ and $o_i = \mathbf{LOW}$ induce the posterior beliefs $n\boldsymbol{x}^*$ and $n\boldsymbol{y}^*$, respectively, regarding the type of action i.

Algorithm 1: Independent Signaling Scheme

Input: Sender payoff vector $\boldsymbol{\xi}$, receiver payoff vector $\boldsymbol{\rho}$, prior distribution $\boldsymbol{q}$

Input: State of nature $\theta \in [m]^n$

Output: An n-dimensional binary signal $\sigma \in \{\mathbf{HIGH}, \mathbf{LOW}\}^n$

1: Compute an optimal solution $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ from linear program (5.3).

2: For each action i independently, set component signal o_i to $\mathbf{HIGH}$ with probability $\frac{x^*_{\theta_i}}{q_{\theta_i}}$ and to $\mathbf{LOW}$ otherwise, where θ_i is the type of action i in the input state θ.

3: Return $\sigma = (o_1, \ldots, o_n)$.

The signaling scheme implemented by Algorithm 1 approximately matches the optimal value of (5.3), as shown in Theorem 5.2.3, assuming the receiver is rational and therefore selects an action with a $\mathbf{HIGH}$ component signal if one exists. We note that the scheme of Algorithm 1, while not a direct scheme as described, can easily be converted into one; specifically, by recommending an action whose component signal is $\mathbf{HIGH}$ when one exists (breaking ties arbitrarily), and recommending an arbitrary action otherwise. Theorem 5.2.3 follows from the fact that $(\boldsymbol{x}^*, \boldsymbol{y}^*)$ is an optimal solution to LP (5.3), the fact that the posterior type distribution of an action i is $n\boldsymbol{x}^*$ when $o_i = \mathbf{HIGH}$ and $n\boldsymbol{y}^*$ when $o_i = \mathbf{LOW}$, and the fact that each component signal is high independently with probability $\frac{1}{n}$. We defer the formal proof to Appendix A.1.3.

Theorem 5.2.3. *Algorithm 1 runs in $poly(m, n)$ time, and serves as a $(1-\frac{1}{e})$-approximate signaling scheme for the Bayesian Persuasion problem with n i.i.d. actions, m types, and nonnegative payoffs.*

Remark 5.2.4. *Algorithm 1 signals independently for each action. This conveys an interesting conceptual message. That is, even though the optimal signaling scheme might induce posterior beliefs which correlate different actions, it is nevertheless true that signaling for each action independently yields an approximately optimal signaling scheme. As a consequence, collaborative persuasion by multiple parties (the senders), each of whom observes the payoff of one or more actions, is a task that can be parallelized, requiring no coordination when actions are identical and independent and only an approximate solution is sought. We leave open the question of whether this is possible when action payoffs are independently but not identically distributed.*

5.2.3 Complexity Barriers to Persuasion with Independent Actions

In this section, we consider optimal persuasion with independent action payoffs as in Section 5.2.2, albeit with action-specific marginal distributions given explicitly. Specifically, for each action i we are given a distribution $q^i \in \Delta_{m_i}$ on m_i types, and each type $j \in [m_i]$ of action i is associated with a sender payoff $\xi^i_j \in \mathbb{R}$ and a receiver payoff $\rho^i_j \in \mathbb{R}$. The positive results

of Section 5.2.2 draw a connection between optimal persuasion in the special case of identically distributed actions and Border's characterization of reduced-form single-item auctions with i.i.d. bidders. One might expect this connection to generalize to the independent non-identical persuasion setting, since Border's theorem extends to single-item auctions with independent non-identical bidders. Surprisingly, we show that this analogy to Border's characterization fails to generalize. We prove the following theorem.

Theorem 5.2.5. *Consider the Bayesian Persuasion problem with independent actions, with action-specific payoff distributions given explicitly. It is $\#P$-hard to compute the optimal expected sender utility.*

Invoking the framework of (Gopalan, Nisan, & Roughgarden, 2015), this rules out a *generalized Border's theorem* for our setting, in the sense defined by (Gopalan et al., 2015), unless the polynomial hierarchy collapses to P^{NP}. We view this result as illustrating some of the important differences between persuasion and mechanism design.

The proof of Theorem 5.2.5 is rather involved. We defer the full proof to Appendix A.2, and only present a sketch here. Our proof starts from the ideas of (Gopalan et al., 2015), who show the #P-hardness for revenue or welfare maximization in several mechanism design problems. In one case, (Gopalan et al., 2015) reduce from the $\#P$-hard problem of computing the *Khintchine constant* of a vector. Our reduction also starts from this problem, but is much more involved:[4] First, we exhibit a polytope which we term the *Khintchine polytope*, and show that computing the Khintchine constant reduces to linear optimization over the Khintchine polytope. Second, we present a reduction from the membership problem for the Khintchine polytope to the computation of optimal sender utility in a particularly-crafted instance of persuasion with independent actions. Invoking the polynomial-time equivalence between membership checking and optimization (see, e.g., (Grötschel, Lovász, & Schrijver, 1988)), we conclude the #P-hardness of our problem. The main technical challenge we overcome is in the second step of our proof: given a vector x which may or may not be in the Khintchine polytope $\mathcal{K}$, we construct a persuasion instance and a threshold T so that points in $\mathcal{K}$ encode signaling schemes, and the optimal sender utility is at least T if and only if $x \in \mathcal{K}$ and the scheme corresponding to x results in sender utility T.

Proof Sketch of Theorem 5.2.5

The *Khintchine problem*, shown to be #P-hard in (Gopalan et al., 2015), is to compute the *Khintchine constant* $K(a)$ of a given vector $a \in \mathbb{R}^n$, defined as $K(a) = \mathbf{E}_{\theta \sim \{\pm 1\}^n}[|\theta \cdot a|]$ where θ

[4] In (Gopalan et al., 2015), Myerson's characterization is used to show that optimal mechanism design in a public project setting directly encodes computation of the Khintchine constant. No analogous direct connection seems to hold here.

is drawn uniformly at random from $\{\pm 1\}^n$. To relate the Khintchine problem to Bayesian persuasion, we begin with a persuasion instance with n *i.i.d.* actions and two action types, which we refer to as *type -1* and *type +1*. The state of nature is a uniform random draw from the set $\{\pm 1\}^n$, with the ith entry specifying the type of action i. We call this instance the *Khintchine-like* persuasion setting. As in Section 5.2.2, we still use the *signature* to capture the payoff-relevant features of a signaling scheme, but we pay special attention to signaling schemes which use only *two* signals, in which case we represent them using a *two-signal signature* of the form $(M^1, M^2) \in \mathbb{R}^{n\times 2} \times \mathbb{R}^{n\times 2}$. The *Khintchine polytope* $\mathcal{K}(n)$ is then defined as the (convex) family of all *realizable* two-signal signatures for the Khintchine-like persuasion problem with an additional constraint: each signal is sent with probability exactly $\frac{1}{2}$. We first prove that general linear optimization over $\mathcal{K}(n)$ is #P-hard by encoding computation of the Khintchine constant as linear optimization over $\mathcal{K}(n)$. In this reduction, the optimal solution in $\mathcal{K}(n)$ is the signature of the two-signal scheme $\varphi(\theta) = sign(\theta \cdot a)$, which signals $+$ and $-$ each with probability $\frac{1}{2}$.

To reduce the membership problem for the Khintchine polytope to optimal Bayesian persuasion, the main challenges come from our restrictions on $\mathcal{K}(n)$, namely to schemes with two signals which are equally probable. Our reduction incorporates three key ideas. The *first* is to design a persuasion instance in which the optimal signaling scheme uses only two signals. The instance we define will have $n + 1$ actions. Action 0 is *special* – it deterministically results in sender utility $\epsilon > 0$ (small enough) and receiver utility 0. The other n actions are *regular*. Action $i > 0$ *independently* results in sender utility $-a_i$ and receiver utility a_i with probability $\frac{1}{2}$ (call this type 1_i), or sender utility $-b_i$ and receiver utility b_i with probability $\frac{1}{2}$ (call this type 2_i), for a_i and b_i to be set later. Note that the sender and receiver utilities are *zero-sum* for both types. Since the special action is deterministic and the probability of its (only) type is 1 in any signal, we can interpret any $(M^1, M^2) \in \mathcal{K}(n)$ as a two-signal signature for our persuasion instance (the row corresponding to the special action 0 is implied). We show that restricting to two-signal schemes is without loss of generality in this persuasion instance. The proof tracks the following intuition: due to the zero-sum nature of regular actions, any additional information regarding regular actions would benefit the receiver and harm the sender. Consequently, the sender does not reveal any information which distinguishes between different regular actions. Formally, we prove that there always exists an optimal signaling scheme with only two signals: one signal recommends the special action, and the other recommends some regular action.

We denote the signal that recommends the special action 0 by σ_+ (indicating that the sender derives positive utility ϵ), and denote the other signal by σ_- (indicating that the sender derives negative utility, as we show). The *second* key idea concerns choosing appropriate values for $\{a_i\}_{i=1}^n, \{b_i\}_{i=1}^n$ for a given two-signature (M^1, M^2) to be tested. We choose these values to satisfy the following two properties: (1) For all regular actions, the signaling scheme implementing

(M^1, M^2) (if it exists) results in the same sender utility -1 (thus receiver utility 1) conditioned on σ_- and the same sender utility 0 conditioned on σ_+; (2) the *maximum possible* expected sender utility from σ_-, i.e., the sender utility conditioned on σ_- multiplied by the probability of σ_-, is $-\frac{1}{2}$. As a result of Property (1), if $(M^1, M^2) \in \mathcal{K}(n)$ then the corresponding signaling scheme φ is persuasive and results in expected sender utility $T = \frac{1}{2}\epsilon - \frac{1}{2}$ (since each signal is sent with probability $\frac{1}{2}$). Property (2) implies that φ results in the maximum possible expected sender utility from σ_-.

We now run into a challenge: the existence of a signaling scheme with expected sender utility $T = \frac{1}{2}\epsilon - \frac{1}{2}$ does not necessarily imply that $(M^1, M^2) \in \mathcal{K}(n)$ if ϵ is large. Our *third* key idea is to set $\epsilon > 0$ "sufficiently small" so that any optimal signaling scheme must result in the maximum possible expected sender utility $-\frac{1}{2}$ from signal σ_- (see Property (2) above). In other words, we must make ϵ so small that the sender prefers to not sacrifice *any* of her payoff from σ_- in order to gain utility from the special action recommended by σ_+. We show that such an ϵ exists with polynomially many bits. We prove its existence by arguing that the polytope of persuasive two-signal signatures has polynomial bit complexity, and therefore an $\epsilon > 0$ that is smaller than the "bit complexity" of the vertices would suffice.

As a result of this choice of ϵ, if the optimal sender utility is precisely $T = \frac{1}{2}\epsilon - \frac{1}{2}$ then we know that signal σ_+ must be sent with probability $\frac{1}{2}$ since the expected sender utility from signal σ_- must be $-\frac{1}{2}$. We show that this, together with the specifically constructed $\{a_i\}_{i=1}^n, \{b_i\}_{i=1}^n$, is sufficient to guarantee that the optimal signaling scheme must implement the given two-signature (M^1, M^2), i.e., $(M^1, M^2) \in \mathcal{K}(n)$. When the optimal optimal sender utility is strictly greater than $\frac{1}{2}\epsilon - \frac{1}{2}$, the optimal signaling scheme does not implement (M^1, M^2), but we show that it can be post-processed into one that does.

5.2.4 An FPTAS for the General Persuasion Problem

We now turn our attention to the Bayesian Persuasion problem when the payoffs of different actions are arbitrarily correlated, and the joint distribution λ is presented as a black-box sampling oracle. We assume that payoffs are normalized to lie in the bounded interval, and prove essentially matching positive and negative results. Our positive result is a fully polynomial-time approximation scheme for optimal persuasion with a bi-criteria guarantee; specifically, we achieve approximate optimality and approximate persuasiveness in the additive sense described in Section 5.1. Our negative results show that such a bi-criteria loss is inevitable in the black box model for information-theoretic reasons.

A Bicriteria FPTAS

Theorem 5.2.6. *Consider the Bayesian Persuasion problem in the black-box oracle model with n actions and payoffs in $[-1, 1]$, and let $\epsilon > 0$ be a parameter. An ϵ-optimal and ϵ-persuasive signaling scheme can be implemented in $\mathrm{poly}(n, \frac{1}{\epsilon})$ time.*

To prove Theorem 5.2.6, we show that a simple Monte-Carlo algorithm implements an approximately optimal and approximately persuasive scheme φ. Notably, our algorithm does not compute a representation of the entire signaling scheme φ as in Section 5.2.2, but rather merely samples its output $\varphi(\theta)$ on a given input θ. At a high level, when given as input a state of nature θ, our algorithm first takes $K = \mathrm{poly}(n, \frac{1}{\epsilon})$ samples from the prior distribution λ which, intuitively, serve to place the true state of nature θ in context. Then the algorithm uses a linear program to compute the optimal ϵ-persuasive scheme $\widetilde{\varphi}$ for the empirical distribution of samples augmented with the input θ. Finally, the algorithm signals as suggested by $\widetilde{\varphi}$ for θ. Details are in Algorithm 2, which we instantiate with $\epsilon > 0$ and $K = \lceil \frac{256n^2}{\epsilon^4} \log(\frac{4n}{\epsilon}) \rceil$.

We note that relaxing persuasiveness is necessary for convergence to the optimal sender utility — we prove this formally in Section 5.2.4. This is why LP (5.4) features relaxed persuasiveness constraints. Instantiating Algorithm 2 with $\epsilon = 0$ results in an exactly persuasive scheme which could be far from the optimal sender utility for any finite number of samples K, as reflected in Lemma 6.

Algorithm 2: Signaling Scheme for a Black Box Distribution

Parameter: $\epsilon \geq 0$

Parameter: Integer $K \geq 0$

Input: Prior distribution λ supported on $[-1, 1]^{2n}$, given by a sampling oracle

Input: State of nature $\theta \in [-1, 1]^{2n}$

Output: Signal $\sigma \in \Sigma$, where $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$.

1: Draw integer ℓ uniformly at random from $\{1, \ldots, K\}$, and denote $\theta_\ell = \theta$.

2: Sample $\theta_1, \ldots, \theta_{\ell-1}, \theta_{\ell+1} \ldots, \theta_K$ independently from λ, and let the multiset $\widetilde{\lambda} = \{\theta_1, \ldots, \theta_K\}$ denote the empirical distribution augmented with the input state $\theta = \theta_\ell$.[5]

3: Solve linear program (5.4) to obtain the signaling scheme $\widetilde{\varphi} : \widetilde{\lambda} \to \Delta(\Sigma)$.

4: Output a sample from $\widetilde{\varphi}(\theta) = \widetilde{\varphi}(\theta_\ell)$.

[5] It is not essential for the algorithm to pick a uniformly random ℓ to set $\theta_l = \theta$. That is, the algorithm also works if we always set $\theta_1 = \theta$. We choose l uniformly at random because this makes θ uniformly distributed in $\widetilde{\lambda}$, conditioned on the samples. This simplifies our proof of Theorem 5.2.6.

$$\begin{array}{lll} \text{maximize} & \sum_{k=1}^{K}\sum_{i=1}^{n}\frac{1}{K}\widetilde{\varphi}(\theta_k,\sigma_i)s_i(\theta_k) & \\ \text{subject to} & \sum_{i=1}^{n}\widetilde{\varphi}(\theta_k,\sigma_i)=1, & \text{for } k\in[K]. \\ & \sum_{k=1}^{K}\frac{1}{K}\widetilde{\varphi}(\theta_k,\sigma_i)r_i(\theta_k)\geq\sum_{k=1}^{K}\frac{1}{K}\widetilde{\varphi}(\theta_k,\sigma_i)(r_j(\theta_k)-\epsilon), & \text{for } i,j\in[n]. \\ & \widetilde{\varphi}(\theta_k,\sigma_i)\geq 0, & \text{for } k\in[K],i\in[n]. \end{array} \tag{5.4}$$

Relaxed Empirical Optimal Signaling Problem

Theorem 5.2.6 follows from three lemmas pertaining to the scheme φ implemented by Algorithm 2.[6] Approximate persuasiveness for λ (Lemma 4) follows from the principle of deferred decisions, linearity of expectations, and the fact that $\widetilde{\varphi}$ is approximately persuasive for the augmented empirical distribution $\widetilde{\lambda}$. A similar argument, also based on the principal of deferred decisions and linearity of expectations, shows that the expected sender utility from our scheme when $\theta \sim \lambda$ equals the expected optimal value of linear program (5.4), as stated in Lemma 5. Finally, we show in Lemma 6 that the optimal value of LP (5.4) is close to the optimal sender utility for λ with high probability, and hence also in expectation, when $K = \text{poly}(n, \frac{1}{\epsilon})$ is chosen appropriately; the proof of this fact invokes standard tail bounds as well as structural properties of linear program (5.4), and exploits the fact that LP (5.4) relaxes the persuasiveness constraint. We prove all three lemmas in Appendix A.3.1. Even though our proof of Lemma 6 is self-contained, we note that it can be shown to follow from (Weinberg, 2014, Theorem 6) with some additional work.

Lemma 4. *Algorithm 2 implements an ϵ-persuasive signaling scheme for prior distribution λ.*

Lemma 5. *Assume $\theta \sim \lambda$, and assume the receiver follows the recommendations of Algorithm 2. The expected sender utility equals the expected optimal value of the linear program* (5.4) *solved in Step 3. Both expectations are taken over the random input θ as well as internal randomness and Monte-Carlo sampling performed by the algorithm.*

Lemma 6. *Let OPT denote the expected sender utility induced by the optimal persuasive signaling scheme for distribution λ. When Algorithm 2 is instantiated with $K \geq \frac{256n^2}{\epsilon^4}\log(\frac{4n}{\epsilon})$ and its input θ is drawn from λ, the expected optimal value of the linear program* (5.4) *solved in Step 3 is at least $OPT - \epsilon$. The expectation is over the random input θ as well as the Monte-Carlo sampling performed by the algorithm.*

Information-Theoretic Barriers

We now show that our bi-criteria FPTAS is close to the best we can hope for: there is no bounded-sample signaling scheme in the black box model which guarantees persuasiveness and

[6] Note that the overall scheme φ implemented by Algorithm 2 should be distinguished from the particular $\widetilde{\varphi}$ for empirical distribution $\widetilde{\lambda}$, which is used to construct $\varphi(\theta)$ for the particular input θ.

c-optimality for any constant $c < 1$, nor is there such an algorithm which guarantees optimality and c-persuasiveness for any $c < \frac{1}{4}$. Formally, we consider algorithms which implement direct signaling schemes. Such an algorithm takes as input a black-box distribution λ supported on $[-1, 1]^{2n}$ and a state of nature $\theta \in [-1, 1]^{2n}$, where n is the number of actions, and outputs a signal $\sigma \in \{\sigma_1, \ldots, \sigma_n\}$ recommending an action. We say such an algorithm is ϵ-persuasive [ϵ-optimal] if for every distribution λ the signaling scheme $\mathcal{A}(\lambda)$ is ϵ-persuasive [ϵ-optimal] for λ. We define the *sample complexity* $SC_{\mathcal{A}}(\lambda, \theta)$ as the expected number of queries made by $\mathcal{A}$ to the blackbox given inputs λ and θ, where the expectation is taken over the randomness inherent in the Monte-Carlo sampling from λ as well as any other internal coins of $\mathcal{A}$. We show that the worst-case sample complexity is not bounded by any function of n and the approximation parameters unless we allow bi-criteria loss in both optimality and persuasiveness. More so, we show a stronger negative result for exactly persuasive algorithms: the average sample complexity over $\theta \sim \lambda$ is also not bounded by a function of n and the suboptimality parameter. Whereas our results imply that we should give up on exact persuasiveness, we leave open the question of whether an optimal and ϵ-persuasive algorithm exists with $\mathrm{poly}(n, \frac{1}{\epsilon})$ average case (but unbounded worst-case) sample complexity.

Theorem 5.2.7. *The following hold for every algorithm $\mathcal{A}$ for Bayesian Persuasion in the black-box model:*

(a) If $\mathcal{A}$ is persuasive and c-optimal for $c < 1$, then for every integer K there is a distribution $\lambda = \lambda(K)$ on 2 actions and 2 states of nature such that $\mathbf{E}_{\theta\sim\lambda}[SC_{\mathcal{A}}(\lambda, \theta)] > K$.

(b) If $\mathcal{A}$ is optimal and c-persuasive for $c < \frac{1}{4}$, then for every integer K there is a distribution $\lambda = \lambda(K)$ on 3 actions and 3 states of nature, and θ in the support of λ, such that $SC_{\mathcal{A}}(\lambda, \theta) > K$.

Our proof of each part of this theorem involves constructing a pair of distributions λ and λ' which are arbitrarily close in statistical distance, but with the property that any algorithm with the postulated guarantees must distinguish between λ and λ'. We defer the proof to Appendix A.3.2.

5.3 Persuading Multiple Receivers

The Bayesian persuasion model examined in Section 5.1 and 5.2 consider the interaction between one sender and one receiver. In this section, we consider a natural generalization in which the sender persuades multiple receivers. We focus on a basic model, first studied in (Arieli & Babichenko, 2016), with binary receiver actions and no externalities. This model generalizes and restricts aspects of the Bayesian persuasion model, and is a fundamental special case for multi-agent persuasion.

5.3.1 A Fundamental Setting: Binary Actions and No Externalities

We adopt the perspective of a *sender* facing n *receivers*. Each receiver has two actions, which we denote by 0 and 1. The receiver's payoff depends only on his own action and a random *state of nature* θ supported on Θ. In particular, we use $u_i(\theta, 1)$ and $u_i(\theta, 0)$ to denote receiver i's utility for action 1 and action 0, respectively, at the state of nature θ; as shorthand, we use $u_i(\theta) = u_i(\theta, 1) - u_i(\theta, 0)$ to denote how much receiver i prefers action 1 over action 0 given the state of nature θ.[7] Note that $u_i(\theta)$ may be negative. The sender's utility (our objective) is a function of all the receivers' actions and the state of nature θ. We use $f_\theta(S)$ to denote the sender's utility when the state of nature is θ and S is the set of receivers who choose action 1. We assume throughout this section that f_θ is a monotone non-decreasing set function for every θ. For convenience in stating our approximation guarantees, we assume without loss of generality that f_θ is normalized so that $f_\theta(\emptyset) = 0$ and $f_\theta(S) \in [0, 1]$ for all $\theta \in \Theta$ and $S \subseteq [n]$.

Like in the Bayesian persuasion (BP) model, θ is drawn from a common prior distribution λ. The sender has access to the realized state of nature and can publicly *commit* to a signaling scheme that reveals to each receiver noisy partial information regarding the state of nature. The main difference from the BP model is that upon observing the realized state θ, the sender will draw a profile of signals $(\sigma_1, \ldots, \sigma_n) \sim \varphi(\theta)$ and send signal σ_i to each receiver i.

Private vs. Public Signaling

A general signaling scheme permits sending different signals to different receivers through a private communication channel — we term these *private signaling schemes* to emphasize this generality. We also study the special case of *public signaling schemes* — these are restricted to a public communication channel, and hence send the same signal to all receivers. We formally define these two signaling models in Sections 5.3.3 and 5.3.5, including the equilibrium concept and the induced sender optimization problem for each. In both cases, we are primarily interested in the optimization problem faced by the sender in step (1), the goal of which is to maximize the sender's expected utility. When φ yields expected sender utility within an additive [multiplicative] ϵ of the best possible, we say it is *ϵ-optimal* [ϵ-approximate] in the additive [multiplicative] sense.

Input Models

We distinguish two input models for describing persuasion instances. The first is the *explicit* model, in which the prior distribution λ is given explicitly as a probability vector. The second is the *sample oracle* model, where Θ and λ are provided implicitly through sample access to λ. In both models, we assume that given a state of nature θ, we can efficiently evaluate $u_i(\theta)$ for

[7] An equivalent presentation is to, w.l.o.g., assume $u_i(\theta, 0) = 0$.

each $i \in [n]$ and $f_\theta(S)$ for each $S \subseteq [n]$. Our analysis will primarily focus on the explicit input model, though we will mention in the context how our results generalize to the implicit input model using techniques from Section 5.2.4.

5.3.2 Technical Preliminaries: Set Functions and Submodularity

Given a finite ground set X, a *set function* is a map $f : 2^X \to \mathbb{R}$. Such a function is *nonnegative* if $f(S) \geq 0$ for all $S \subseteq X$, *monotone non-decreasing* (or *monotone* for short) if $f(S) \leq f(T)$ for all $S \subseteq T$. Most importantly, f is *submodular* if for any $S, T \subseteq X$, we have $f(S \cup T) + f(S \cap T) \leq f(S) + f(T)$. Submodular functions are widely used to model utilities for a set of items.

We also consider continuous functions G from the solid hypercube $[0,1]^X$ to the real numbers. Such a function is *nonnegative* if $G(x) \geq 0$ for all $x \in [0,1]^X$, *monotone non-decreasing* (or *monotone* for short) if $G(x) \leq G(y)$ whenever $x \preceq y$ (coordinate wise), and *smooth submodular* (in the sense of (Calinescu, Chekuri, Pál, & Vondrák, 2011)) if its second partial derivatives exist and are non-positive everywhere.

The Multilinear Extension of a Set Function. Given any set function $f : 2^X \to \mathbb{R}$, the *multilinear extension* of f is the continuous function $F : [0,1]^X \to \mathbb{R}$ defined as follows:

$$F(x) = \sum_{S \subseteq X} f(S) \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i), \tag{5.5}$$

Notice that $F(x)$ can be viewed as the expectation of $f(S)$ when the random set S independently includes each element i with probability x_i. In particular, let p_x^I denote the *independent distribution* with marginals x, defined by $p_x^I(S) = \prod_{i \in S} x_i \prod_{i \notin S} (1 - x_i)$. Then $F(x) = \mathbf{E}_{S \sim p_x^I} f(S)$. If f is nonnegative/monotone then so is F. Moreover, if f is submodular then F is smooth submodular. For our results, we will need to maximize $F(x)$ subject to a set of linear constraints on x. This problem is NP-hard in general, yet can be approximated by the *continuous greedy process* of (Calinescu et al., 2011) for fairly general families of constraints. Note that though we cannot exactly evaluate $F(x)$ in polynomial time, it is sufficient to approximate $F(x)$ within a good precision in order to apply the continuous greedy process. By an additive FPTAS evaluation oracle for F, we mean an algorithm that evaluates $F(x)$ within additive error ϵ in $\mathrm{poly}(n, \frac{1}{\epsilon})$ time.

Theorem 5.3.1 (Adapted form (Calinescu et al., 2011))**.** *Let $F : [0,1]^n \to [0,1]$ be a nonnegative, monotone, smooth submodular function. Let $\mathcal{P} \subseteq [0,1]^n$ be a down-monotone polytope*[8]*, specified explicitly by its linear constraints. Given an additive FPTAS evaluation oracle*

[8] *A polytope $\mathcal{P} \subseteq \mathbb{R}_+^n$ is called* down-monotone *if for all $x, y \in \mathbb{R}_+^n$, if $y \in P$ and $x \preceq y$ (coordinate-wise) then $x \in P$.*

for F, there is a $\mathrm{poly}(n, \frac{1}{\epsilon})$ *time algorithm that outputs* $\overline{x} \in \mathcal{P}$ *such that* $F(\overline{x}) \geq (1-\frac{1}{e})OPT-\epsilon$*, where* $OPT = \max_{x\in\mathcal{P}} F(x)$.

Correlation Gap. A general definition of the correlation gap can be found in (Agrawal, Ding, Saberi, & Ye, 2010). For our results, the following simple definition will suffice. Specifically, for any $x \in [0,1]^X$, let $D(x)$ be the set of all distributions p over 2^X with fixed marginal probability $\mathbf{Pr}_{S\sim p}(i \in S) = x_i$ for all i. Let p_x^I, as defined above, be the independent distribution with marginal probabilities x. Note that $p_x^I \in D(x)$. For any set function $f(S)$, the correlation gap κ is defined as follows:

$$\kappa = \max_{x\in[0,1]^X} \max_{p\in D(x)} \frac{\mathbf{E}_{S\sim p} f(S)}{\mathbf{E}_{S\sim p_x^I} f(S)}. \tag{5.6}$$

Loosely speaking, the correlation gap upper bounds the "loss" of the expected function value over a random set by ignoring the correlation in the randomness.

Theorem 5.3.2. *(Agrawal et al., 2010) The correlation gap* κ *is upper bounded by* $\frac{e}{e-1}$ *for any non-negative monotone non-decreasing submodular function.*

5.3.3 Optimal Private Persuasion and Its Complexity Characterization

A *private signaling scheme* φ is a randomized map from the set of states of nature Θ to a set of *signal profiles* $\Sigma = \Sigma_1 \times \Sigma_2 \times \cdots \Sigma_n$, where Σ_i is the *signal set* of receiver i. We use $\varphi(\theta, \sigma)$ to denote the probability of selecting the signal profile $\sigma = (\sigma_1, \ldots, \sigma_n) \in \Sigma$ given a state of nature θ. Therefore, $\sum_{\sigma\in\Sigma} \varphi(\theta, \sigma) = 1$ for every θ. With some abuse of notation, we use $\varphi(\theta)$ to denote the random signal profile selected by the scheme φ given the state θ. Moreover, for each $\theta \in \Theta$, $i \in [n]$, and $\sigma_i \in \Sigma_i$, we use $\varphi_i(\theta, \sigma_i) = \mathbf{Pr}[\varphi_i(\theta) = \sigma_i]$ to denote the marginal probability that receiver i receives signal σ_i in state θ. An algorithm *implements* a signaling scheme φ if it takes as input a state of nature θ, and samples the random variable $\varphi(\theta)$.

Given a signaling scheme φ, each signal $\sigma_i \in \Sigma_i$ for receiver i is realized with probability $\mathbf{Pr}(\sigma_i) = \sum_{\theta\in\Theta} \lambda(\theta)\varphi_i(\theta, \sigma_i)$. Upon receiving σ_i, receiver i — like the receiver in the BP model — performs a Bayesian update and infers a posterior belief over the state of nature, as follows: the realized state is θ with posterior probability $\lambda(\theta)\varphi_i(\theta, \sigma_i)/\mathbf{Pr}(\sigma_i)$. Receiver i then takes the action maximizing his posterior expected utility. In case of indifference, we assume ties are broken in favor of the sender (i.e., in favor of action 1). Therefore, receiver i chooses action 1 if

$$\frac{1}{\mathbf{Pr}(\sigma_i)} \sum_{\theta\in\Theta} \lambda(\theta)\varphi_i(\theta, \sigma_i)u_i(\theta, 1) \geq \frac{1}{\mathbf{Pr}(\sigma_i)} \sum_{\theta\in\Theta} \lambda(\theta)\varphi_i(\theta, \sigma_i)u_i(\theta, 0),$$

or equivalently

$$\sum_{\theta\in\Theta} \lambda(\theta)\varphi_i(\theta, \sigma_i)u_i(\theta) \geq 0,$$

where $u_i(\theta) = u_i(\theta, 1) - u_i(\theta, 0)$.

Like in the BP model, a revelation-principle style argument shows that there exist an optimal private signaling scheme which is *direct* and *persuasive* (Kamenica & Gentzkow, 2011; Arieli & Babichenko, 2016). By *direct* we mean that signals correspond to actions — in our setting $\Sigma_i = \{0, 1\}$ for each receiver i — and can be interpreted as action recommendations. A direct scheme is *persuasive* if the strategy profile where all receivers follow their recommendations forms an equilibrium of the resulting Bayesian game. Due to the absence of inter-receiver externalities in our setting, such an equilibrium will necessarily also satisfy the stronger property of being a dominant-strategy equilibrium — i.e., each receiver i maximizes his posterior expected utility by following the recommendation, regardless of whether other receivers follow their recommendations.

When designing private signaling schemes, we restrict attention (without loss) to direct and persuasive schemes. Here, a signal profile can be equivalently viewed as a set $S \subseteq [n]$ of receivers — namely, the set of receivers who are recommended action 1. Using this alternative representation, a scheme can be specified by variables $\varphi(\theta, S)$ for all $\theta \in \Theta, S \subseteq [n]$. We can now encode the sender's optimization problem of computing the optimal scheme using the following exponentially large linear program; note the use of auxiliary variables $x_{\theta,i}$ to denote the marginal probability of recommending action 1 to receiver i in state θ.

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta} \lambda(\theta) \sum_{S\subseteq[n]} \varphi(\theta, S) f_\theta(S) & \\ \text{subject to} & \sum_{S:i\in S} \varphi(\theta, S) = x_{\theta,i}, & \text{for } i \in [n], \theta \in \Theta. \\ & \sum_{\theta\in\Theta} \lambda(\theta) x_{\theta,i} u_i(\theta) \geq 0, & \text{for } i = 1, \ldots, n. \\ & \sum_{S\subseteq[n]} \varphi(\theta, S) = 1, & \text{for } \theta \in \Theta. \\ & \varphi(\theta, S) \geq 0, & \text{for } \theta \in \Theta; S \subseteq [n]. \end{array} \tag{5.7}$$

The second set of constraints in LP (5.7) are *persuasiveness constraints*, and state that each receiver i should maximize his utility by taking action 1 whenever action 1 is recommended. Note that the persuasiveness constraints for action 0, which can be written as $\sum_{\theta\in\Theta} \lambda(\theta)(1 - x_{\theta,i})u_i(\theta) \leq 0$ for each $i \in [n]$, are intentionally omitted from this LP. This omission is without loss when f_θ is a non-decreasing set function for each θ: any solution to the LP in which a receiver prefers action 1 when recommended action 0 can be improved by always recommending action 1 to that receiver.

Since the size of LP (5.7) is exponential in the input size of the problem, it is not clear whether we can solve the problem in time polynomial in the input size. Next, we study the complexity of optimal private persuasion. In particular, we relate the computational complexity of private persuasion to the complexity of maximizing the sender's objective function, and show that the optimal private signaling scheme can be computed efficiently for a broad class of sender

objectives. Let $\mathcal{F}$ denote any collection of monotone set functions. We use $\mathcal{I}(\mathcal{F})$ to denote the class of all persuasion instances in our model in which the sender utility function f_θ is in $\mathcal{F}$ for all states of nature θ. We restrict attention to the explicit input model for most of this discussion, though discuss how to extend our results to the sample oracle model, modulo an arbitrarily small additive loss in both the sender's objective and the persuasiveness constraints, at the end of this section.

The following theorem establishes the polynomial-time equivalence between computing the optimal private signaling scheme and the problem of maximizing the objective function plus an additive function. Note that although the number of variables in LP (5.7) is exponential in the number of receivers, a vertex optimal solution of this LP is supported on $O(n|\Theta|)$ variables.

Theorem 5.3.3. *Let $\mathcal{F}$ be any collection of monotone set functions. There is a polynomial-time algorithm which computes the optimal private signaling scheme given any instance in $\mathcal{I}(\mathcal{F})$ if and only if there is a polynomial time algorithm for maximizing $f(S) + \sum_{i \in S} w_i$ given any $f \in \mathcal{F}$ and any set of weights $w_i \in \mathbb{R}$.*

Proof. We first reduce optimal private signaling to maximizing the objective function plus an additive function, via linear programming duality. Consider the following dual program of LP (5.7) with variables $w_{\theta,i}, \alpha_i, y_\theta$.

$$\begin{array}{lll} \text{minimize} & \sum_{\theta \in \Theta} y_\theta & \\ \text{subject to} & \sum_{i \in S} w_{\theta,i} + y_\theta \geq \lambda(\theta) f_\theta(S), & \text{for } S \subseteq [n], \theta \in \Theta. \\ & w_{\theta,i} + \alpha_i \lambda(\theta) u_i(\theta) = 0, & \text{for } i = 1, \ldots, n. \\ & \alpha_i \geq 0, & \text{for } i \in [n]. \end{array} \tag{5.8}$$

We can obtain a separation oracle for LP (5.8) given an algorithm for maximizing $f_\theta(S)$ plus an additive function. Given any variables $w_{\theta,i}, \alpha_i, y_\theta$, separation over the first set of constraints reduces to maximizing the set function $g_\theta(S) = f_\theta(S) - \frac{1}{\lambda(\theta)} \sum_{i \in S} w_{\theta,i}$ for each $\theta \in \Theta$. The other constraints can be checked directly in linear time. Given the resulting separation oracle, we can use the Ellipsoid method to obtain a vertex optimal solution to both LP (5.8) and its dual LP (5.7) in polynomial time (Grötschel et al., 1988).

We now prove the converse. Namely, we construct a polynomial-time Turing reduction from the problem of maximizing f plus an additive function to a private signaling problem in $\mathcal{I}(\mathcal{F})$. At a high level, we first reduce the set function maximization problem to a certain linear program, and then prove that solving the dual of the LP reduces to optimal private signaling for a set of particularly constructed instances in $\mathcal{I}(\mathcal{F})$.

Given $f \in F$ and weights w, our reduction concerns the following linear program, parameterized by $\mathbf{a} = (a_1, \ldots, a_n)$ and b, with variables $\mathbf{z} = (z_1, \ldots, z_n)$ and v.

$$\begin{array}{ll} \text{minimize} & \sum_{i \in [n]} a_i z_i + bv \\ \text{subject to} & \sum_{i \in S} z_i + v \geq f(S), \quad \text{for } S \subseteq [n]. \end{array} \tag{5.9}$$

Let $\mathcal{P}$ denote the feasible region of LP (5.9). As the first step of our reduction, we reduce maximizing the set function $g_w(S) = f(S) + \sum_{i \in S} w_i$ to the separation problem for $\mathcal{P}$. Let $z_i = -w_i$ for each i. Notice that $(\mathbf{z}, v)$ is feasible (i.e., in $\mathcal{P}$) if and only if $v \geq \max_{S \subseteq [n]} f(S) - \sum_{i \in S} z_i$. Therefore, we can binary search for a value $\widetilde{v}$ such that $(\mathbf{z}, \widetilde{v})$ is almost feasible, but not quite. More precisely, let B denote the bit complexity of the $f(S)$'s and the w_i's. Then binary search returns the exact optimal value of the set function maximization problem after $\mathcal{O}(B)$ steps. We then set $\widetilde{v}$ to equal that value minus 2^{-B}. Feeding $(\mathbf{z}, \widetilde{v})$ to the separation oracle, we obtain a violated constraint which must correspond to the maximizer of $f(S) + \sum_{i \in S} w_i$.

As the second step of our reduction, we reduce the separation problem for $\mathcal{P}$ to solving LP (5.9) for every choice of objective coefficients $\mathbf{a}$ and b. This polynomial-time Turing reduction follows from the equivalence of separation and optimization (Grötschel et al., 1988).

Third, we reduce solving LP (5.9) for arbitrary $\mathbf{a}$ and b to the special case where $\mathbf{a} \in [0,1]^n$ and $b = 1$. The reduction involves a case analysis. (a) If any of the objective coefficients are negative, then the fact that $\mathcal{P}$ is upwards closed implies that LP (5.9) is unbounded. (b) If $b = 0$ and $a_i > 0$ for some i, then the LP is unbounded since we can make v arbitrarily small and z_i arbitrarily large. Normalizing by dividing by b, we have reduced the problem to the case when $b = 1$ and $a \succeq 0$ (coordinate-wise). (c) Now suppose that $a_i > 1 = b$ for some i; the LP is unbounded by making z_i arbitrarily small and v arbitrarily large. This analysis leaves the case of $b = 1$ and $\mathbf{a} \in [0,1]^n$.

Fourth, we reduce LP (5.9) with parameters $\mathbf{a} \in [0,1]^n$ and $b = 1$ to its dual shown below, with variables p_S for $S \subseteq [n]$.

$$\begin{array}{ll} \text{maximize} & \sum_{S \subseteq [n]} p_S f(S) \\ \text{subject to} & \sum_{S : i \in S} p_S \leq a_i, \quad \text{for } i \in [n]. \\ & \sum_{S \subseteq [n]} p_S = 1 \\ & p_S \geq 0, \qquad\qquad \text{for } S \subseteq [n]. \end{array} \tag{5.10}$$

We note that LP (5.10) is not the standard dual of LP (5.9). In particular, the first set of constraints are inequality rather than equality constraints. It is easy to see that LP (5.10) is equivalent to the standard dual when f is monotone non-decreasing, and that an optimal solution to one of the two duals can be easily converted to an optimal solution of the other.

The fifth and final step of our reduction reduces LP (5.10) to a private signaling problem in $\mathcal{I}(\mathcal{F})$. There are n receivers and two states of nature θ_0, θ_1 with $\lambda(\theta_0) = \lambda(\theta_1) = 1/2$. Define

$u_i(\theta_1) = 1$ and $u_i(\theta_0) = -\frac{1}{a_i}$ ($-\infty$ if $a_i = 0$) for all i. The sender's utility function satisfies $f_{\theta_1} = f_{\theta_0} = f$. Let φ^* be an optimal signaling scheme, in particular an optimal solution to the instantiation of LP (5.7) for our instance. Note that all receivers prefer action 1 in state θ_1; therefore, φ^* can be weakly improved, without violating the persuasiveness constraints, by modifying it to always recommend action 1 to all receivers when in state θ_1. After this modification, φ^* is an optimal solution to the following LP, which optimizes over all signaling schemes satisfying $\varphi(\theta_1, [n]) = 1$.

$$\begin{array}{lll} \text{maximize} & \frac{1}{2}f([n]) + \frac{1}{2}\sum_{S\subseteq[n]} \varphi(\theta_0, S)f(S) & \\ \text{subject to} & \sum_{S:i\in S} \varphi(\theta_0, S) = x_{\theta_0,i}, & \text{for } i \in [n]. \\ & x_{\theta_0,i} \le a_i, & \text{for } i = 1, \ldots, n. \\ & \sum_{S\subseteq[n]} \varphi(\theta_0, S) = 1 & \\ & \varphi(\theta_0, S) \ge 0, & \text{for } \theta \in \Theta; S \subseteq [n]. \end{array} \tag{5.11}$$

It is now easy to see that setting $p_S = \varphi^*(\theta_0, S)$ yields an optimal solution to LP (5.10) □

As an immediate corollary of Theorem 5.3.3, the optimal private signaling scheme can be computed efficiently when the sender's objective function is supermodular or anonymous. Recall that a set function $f : 2^{[n]} \to \mathbb{R}$ is anonymous if there exists a function $g : \mathbb{Z} \to \mathbb{R}$ such that $f(S) = g(|S|)$.

Corollary 1. *There is a polynomial-time algorithm for computing the optimal private signaling scheme when the sender objective functions are either supermodular or anonymous.*

Proof. Since a supermodular function plus an additive function is still supermodular, and the problem of unconstrained supermodular maximization can be solved in polynomial time , Theorem 5.3.3 implies that the optimal private signaling scheme can also be computed in polynomial time. As for anonymous objectives, there is a simple algorithm for maximizing an anonymous set function plus an additive function. In particular, consider the problem of maximizing $f(S) + \sum_{i\in S} w_i$ where $f(S) = g(|S|)$. Observe that fixing $|S| = k$, the optimal set S_k corresponds to the k highest-weight elements in w. Enumerating all k and choosing the best S_k yields the optimal set. □

Finally, we make two remarks on Theorem 5.3.3, particularly on the reduction from optimal private signaling to set function maximization. First, the assumption of monotonicity is not necessary to the reduction from signaling to optimization. In other words, even without the monotonicity assumption for the sender's objective function, one can still efficiently compute the optimal private signaling scheme for instances in $\mathcal{I}(\mathcal{F})$ given access to an oracle for maximizing $f(S) + \sum_{i\in S} w_i$ for any $f \in \mathcal{F}$ and weight vector w. This can be verified by adding the persuasiveness constraints for action 0 back to LP (5.7) and examining the corresponding dual, which

has similar structure to LP (5.8). We omit the details here. Consequently, Corollary 1 applies to non-monotone supermodular or anonymous functions as well.

Second, our reduction assumes that the prior distribution λ over the state of nature is explicitly given. This can be generalized to the sample oracle model. In particular, when our only access to λ is through random sampling, we can implement an ϵ-optimal and ϵ-persuasive private signaling scheme in $poly(n, \frac{1}{\epsilon})$ time using the idea in Section 5.2.4. (assuming $u_i(\theta) \in [-1, 1]$). The algorithm is as follows: given any input state θ, we first take $poly(n, \frac{1}{\epsilon})$ samples from λ, and then solve LP (5.7) on the empirical distribution of the samples plus θ, with relaxed (by ϵ) persuasiveness constraints. Finally, we signal for θ as the solution to the LP suggests. The analysis of this algorithm is very similar to that in Section 5.2.4, and is omitted here. Moreover, the bi-criteria loss is inevitable in this oracle model due to information theoretic reasons.

5.3.4 Private Persuasion with Submodular Objectives

Theorem 5.3.3 relates the exact computation of the optimal private signaling scheme to exact maximization of (a variant of) the set function $f(S)$. One natural question is what if exactly maximizing the set function $f(S)$ is intractable and we can only obtain an approximate solution efficiently. An important case of such a scenario is when $f(S)$ is submodular.

To answer this question, we consider optimal private signaling for submodular sender objectives in this section, and show that there is a polynomial time $(1 - \frac{1}{e})$-approximation scheme, modulo an additive loss of ϵ. This is almost the best possible: (Babichenko & Barman, 2017) show that even in the special case of two states of nature, it is NP-hard to approximate the optimal private signaling scheme within a factor better than $(1 - \frac{1}{e})$ for monotone submodular sender objectives.

Theorem 5.3.4. *Consider private signaling with monotone submodular sender objectives. Let OPT denote the optimal sender utility. For any $\epsilon > 0$, a private signaling scheme achieving expected sender utility at least $(1 - \frac{1}{e})OPT - \epsilon$ can be implemented in $\mathrm{poly}(n, |\Theta|, \frac{1}{\epsilon})$ time.*

The main technical challenge in proving Theorem 5.3.4 is that a private signaling scheme may have exponentially large support, as apparent from linear program (5.7). To overcome this difficulty, we prove a structural characterization of (approximately) optimal persuasive private schemes, i.e., solutions to LP (5.7). Roughly speaking, we show that LP (5.7) always has an approximately optimal solution with polynomial-sized support and nicely structured distributions. This greatly narrows down the solution space we need to search over. Recall that for any θ, $\varphi(\theta)$ is a random variable supported on $2^{[n]}$. We say $\varphi(\theta)$ is *K-uniform* if it follows a uniform distribution on a multiset of size K. The following lemma exhibits a structural property regarding

(approximately) optimal solutions to LP (5.7). Notably, this property only depends on monotonicity of the sender's objective functions and does not depend on submodularity. Its proof is postponed to the end of this section.

Lemma 7. *Let f_θ be monotone for each θ. For any $\epsilon > 0$, there exists an ϵ-optimal persuasive private signaling scheme $\overline{\varphi}$ such that $\overline{\varphi}(\theta)$ is K-uniform for every θ, where $K = \frac{108n\log(2n|\Theta|)}{\epsilon^3}$.*

By Lemma 7, we can, without much loss, restrict our design of $\varphi(\theta)$ to the special class of K-uniform distributions. Note that a K-uniform distribution $\varphi(\theta)$ can be described by variables $x^j_{\theta,i} \in \{0,1\}$ for $i \in [n], j \in [K]$, where $x^j_{\theta,i}$ denotes the recommended action to receiver i in the j'th profile in the support of $\varphi(\theta)$. Relaxing our variables to lie in $[0,1]$, this leads to optimization problem (5.12), where $F_\theta(x) = \sum_{S\subseteq[n]} f_\theta(S)\prod_{i\in S} x_i \prod_{i\notin S}(1-x_i)$ is the multi-linear extension of f_θ.

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta} \frac{\lambda(\theta)}{K} \sum_{j=1}^{K} F_\theta(x^j_\theta) & \\ \text{subject to} & \sum_{\theta\in\Theta} \frac{\lambda(\theta)}{K} \sum_{j=1}^{K} x^j_{\theta,i} u_i(\theta) \geq 0, & \text{for } i = 1,\ldots,n. \\ & 0 \leq x^j_{\theta,i} \leq 1, & \text{for } i = 1,\ldots,n; \theta \in \Theta. \end{array} \tag{5.12}$$

At a high level, our algorithm first approximately solves Program (5.12) and then signals according to its solution. Details are in Algorithm 3, which we instantiate with $\epsilon > 0$ and $K = \frac{108n\log(2n|\Theta|)}{\epsilon^3}$. Since $F_\theta(x) = \mathbf{E}_{S\sim p^I_x} f(S)$ where p^I_x is the independent distribution over $2^{[n]}$ with marginal probability x, the expected sender utility induced by the signaling scheme in Algorithm 3 is precisely the objective value of Program (5.12) at the obtained solution. Theorem 5.3.4 then follows from two claims: 1. The optimal objective value of Program (5.12) is ϵ-close to the optimal sender utility (Claim 1); 2. The continuous greedy process (Calinescu et al., 2011) can be applied to Program (5.12) to efficiently compute a $(1-1/e)$-approximate solution, modulo a small additive loss (Claim 2). We remark that Theorem 5.3.4 can be generalized to the sample oracle model, but with an additional ϵ-loss in persuasiveness constraints (assuming $u_i(\theta) \in [-1,1]$), using the idea from Section 5.2.4.

Claim 1. *When $K = \frac{108n\log(2n|\Theta|)}{\epsilon^3}$, the optimal objective value of Program* (5.12) *is at least $OPT - \epsilon$, where OPT is the optimal sender utility in private signaling.*

Proof. By Lemma 7, there exists a private signaling scheme $\overline{\varphi}$ such that: (i) $\overline{\varphi}$ achieves sender utility at least $OPT - \epsilon$; (ii) for each θ, there exists K sets $S^1_\theta, \ldots, S^K_\theta \subseteq [n]$ such that $\overline{\varphi}_\theta$ is a uniform distribution over $\{S^1_\theta, \ldots, S^K_\theta\}$. Utilizing $\overline{\varphi}$, we can construct a feasible solution $\overline{x}$ to Program (5.12) with objective value at least $OPT - \epsilon$. In particular, let $\overline{x}^j_\theta \in \{0,1\}^n$ be the indicator vector of the set S^j_θ, formally defined as follows: $\overline{x}^j_{\theta,i} = 1$ if and only if $i \in S^j_\theta$. By

Algorithm 3: Private Signaling Scheme for Submodular Sender Objectives

Parameter: $\epsilon > 0$
Input: Prior distribution λ supported on Θ
Input: $u_i(\theta)$'s and value oracle access to the sender utility $f_\theta(S)$
Input: State of nature θ
Output: A set $S \subseteq [n]$ indicating the set of receivers who will be recommended action 1.

1: Approximately solve Program (5.12). Let $\{\widetilde{x}^j_{\theta,i}\}_{\theta\in\Theta, i\in[n], j\in[K]}$ be the returned solution.
2: Choose j from $[K]$ uniformly at random; For each receiver i, add i to S independently with probability $\widetilde{x}^j_{\theta,i}$.
3: Return S.

referring to the feasibility of $\overline{\varphi}$ for LP (5.7), it is easy to check that $\overline{x}^j_{\theta,i}$'s are feasible for Program (5.12). Moreover, since $F_\theta(\overline{x}^j_\theta) = f_\theta(S^j_\theta)$, the objective value of Program (5.12) at the solution $\overline{x}$ equals the objective value of Program (5.7) at the solution $\overline{\varphi}$, which is at least $OPT - \epsilon$. Therefore, the optimal objective value of Program (5.12) is at least $OPT - \epsilon$, as desired. □

Claim 2. *There is an algorithm that runs in* $\mathrm{poly}(n, |\Theta|, K, \frac{1}{\epsilon})$ *time and computes a* $(1 - 1/e)$*-approximate solution, modulo an additive loss of* ϵ/e*, to Program* (5.12)*.*

Proof. The objective function of Program (5.12) is a linear combination, with non-negative coefficients, of multilinear extensions of monotone submodular functions, and thus is smooth, monotone and submodular. Moreover, the function value can be evaluated within error ϵ by $\mathrm{poly}(n, \frac{1}{\epsilon})$ random samples, and thus in $\mathrm{poly}(n, \frac{1}{\epsilon})$ time. To apply Theorem 5.3.1, we only need to prove that the feasible region is a down-monotone polytope. Observe that there always exists an optimal solution to Program (5.12) such that $x_{\theta,i} = 1$ for any θ, i such that $u_i(\theta) \geq 0$. Therefore, w.l.o.g., we can pre-set these variables to be 1 and view the program as an optimization problem over $x_{\theta,i}$'s for all θ, i such that $u_i(\theta) < 0$. It is easy to verify that these $x_{\theta,i}$'s form a down-monotone polytope determined by polynomially many constraints, as desired. □

Proof of Lemma 7

Our proof is based on the probabilistic method. Recall that the optimal private signaling scheme can be computed by solving the exponentially large LP (5.7). Roughly speaking, given any optimal private scheme φ^*, we will take polynomially many samples from $\varphi^*(\theta)$ for each θ, and prove that with strictly positive probability the corresponding empirical distributions form a solution to LP (5.7) that is close to optimality. However, the sampling approach usually suffers from ϵ-loss in both the objective and persuasiveness constraints. It turns out that the ϵ-loss in persuasiveness constraints can be avoided in our setting with carefully designed pre-processing steps.

At a high level, to get rid of the ϵ-loss in persuasiveness constraints, there are two main technical barriers. The first is to handle the estimation error in the receiver's utilities, which is inevitable due to sampling. We address this by adjusting φ^* to strengthen the persuasiveness constraints so that a small estimation error still preserves the original persuasiveness constraints. The second barrier arises when some $x^*_{\theta,i}$'s are smaller than inverse polynomial of the precision ϵ. Then $poly(\frac{1}{\epsilon})$ samples cannot guarantee a good multiplicative estimate of $x^*_{\theta,i}$. We deal with this issue by making the "honest" recommendation, i.e., action 0, in these cases, and show that such a modification does not cause much loss in our objective.

We first introduce some convenient notations. For any receiver i, let $\Theta_i^+ = \{\theta : u_i(\theta) \geq 0\}$ be the set of states in which receiver i (weakly) prefers action 1; similarly, $\Theta_i^- = \{\theta : u_i(\theta) < 0\}$ is the set of states in which receiver i strictly prefers action 0. For any state of nature θ, let $I_\theta^+ = \{i : u_i(\theta) \geq 0\}$ be the set of receivers who (weakly) prefer action 1 in state θ. It is convenient to think of $\{\Theta_i^+\}_{i\in[n]}$ and $\{I_\theta^+\}_{\theta\in\Theta}$ as two different partitions of the set $\{(\theta, i) : u_i(\theta) \geq 0\}$.

Observe that by monotonicity there always exists an optimal signaling scheme φ^* such that $x^*_{\theta,i} = 1$ for every $\theta \in \Theta_i^+$. Let φ^* be such an optimal signaling scheme and OPT denote the optimal sender utility. We now adjust the scheme φ^* without degrading the objective value by much but such that the scheme is more suitable for applying concentration bounds for our probabilistic argument.

Adjustment 1: Always Recommend Action 0 When $x^*_{\theta,i} < \frac{\epsilon}{3n}$

Note that $x^*_{\theta,i} < \frac{\epsilon}{3n}$ only when $\theta \in \Theta_i^-$, i.e., action 0 is the best action for receiver i conditioned on θ. We first adjust φ^* to obtain a new scheme $\widetilde{\varphi}$, as follows: $\widetilde{\varphi}$ is the same as φ^* except that for every θ, i such that $x^*_{\theta,i} < \frac{\epsilon}{3n}$, $\widetilde{\varphi}$ always recommends action 0 to receiver i given the state of nature θ. As a result, $\widetilde{x}_{\theta,i}$ equals $x^*_{\theta,i}$ whenever $x^*_{\theta,i} \geq \frac{\epsilon}{3n}$ and equals 0 otherwise. Note that the signaling scheme still satisfies the persuasiveness constraints.

Naturally, each adjustment above, corresponding to θ, i satisfying $x^*_{\theta,i} < \frac{\epsilon}{3n}$, could decrease the objective value since the marginal probability of recommending action 1 decreases. Nevertheless, this loss, denoted as $L(\theta, i)$, can be properly bounded as follows:

$$\begin{aligned} L(\theta,i) &= \lambda(\theta)\cdot\left[\sum_{S:i\in S}\varphi^*(\theta,S)f_\theta(S) - \sum_{S:i\in S}\varphi^*(\theta,S)f_\theta(S\setminus\{i\})\right] \\ &\leq \lambda(\theta)\cdot\left[\sum_{S:i\in S}\varphi^*(\theta,S)\right] \\ &= \lambda(\theta)x^*_{\theta,i} \leq \frac{\lambda(\theta)\epsilon}{3n}. \end{aligned}$$

As a result, the aggregated loss of all the adjustments made in this step can be upper bounded by $\sum_{\theta\in\Theta}\sum_{i=1}^{n}\frac{\lambda(\theta)\epsilon}{3n}=\frac{\epsilon}{3}$. That is, the objective value of $\widetilde{\varphi}$ is at least $OPT-\frac{\epsilon}{3}$.

Adjustment 2: Strengthen the Persuasiveness Constraints by Scaling Down $x_{\theta,i}$'s

We now strengthen the persuasiveness constraints by further adjusting the $\widetilde{\varphi}$ obtained above so that a small estimation error due to sampling will still maintain the original persuasiveness constraints. For any θ, we define $\varphi'(\theta,S)=\frac{3}{3+\epsilon}\widetilde{\varphi}(\theta,S)$ for all $S\neq I_\theta^+$, and define $\varphi'(\theta,I_\theta^+)=1-\sum_{S\neq I_\theta^+}\varphi'(\theta,S)$. Obviously, φ'_θ is still a distribution over $2^{[n]}$. We claim that $x'_{\theta,i}=\mathbf{E}_{S\sim\varphi'_\theta}\mathbb{I}(i\in S)=1$ whenever $\widetilde{x}_{\theta,i}=1$, i.e., $\theta\in\Theta_i^+$. That is, given state θ, any receiver $i\in I_\theta^+$ will still aways be recommended action 1. This is because, to construct φ'_θ, we moved some probability mass from all other sets S to the set I_θ^+; therefore the marginal probability of recommending action 1 to any receiver $i\in I_\theta^+$ will not decrease. However, this marginal probability is originally 1 in the solution of $\widetilde{\varphi}$. Therefore, $x'_{\theta,i}$ still equals 1 for any $i\in I_\theta^+$, or equivalently, for any $\theta\in\Theta_i^+$. Similarly, we also have $x'_{\theta,i}=0$ whenever $\widetilde{x}_{\theta,i}=0$.

Let $Val(\varphi)$ denote the objective value of a scheme φ. We claim that $Val(\varphi')\geq OPT-\frac{2\epsilon}{3}$ and φ' satisfies $x'_{\theta,i}=\frac{3}{3+\epsilon}\widetilde{x}_{\theta,i}$ for every $\theta\in\Theta_i^-$. For any $i\in[n],\theta\in\Theta_i^-$ (which means $i\notin I_\theta^+$), we have

$$x'_{\theta,i}=\sum_{S:i\in S}\varphi'(\theta,S)=\frac{3}{3+\epsilon}\sum_{S:i\in S}\widetilde{\varphi}(\theta,S)=\frac{3}{3+\epsilon}\widetilde{x}_{\theta,i},$$

since the summation excludes the term $\varphi'(\theta,I_\theta^+)$. We now prove the guarantee of the objective value. Observe that $\varphi'(\theta,I_\theta^+)\geq\frac{3}{3+\epsilon}\widetilde{\varphi}(\theta,I_\theta^+)$ also holds in our construction. Therefore, we have

$$\begin{aligned}
Val(\varphi') &= \sum_{\theta\in\Theta}\lambda(\theta)\sum_{S\subseteq[n]}\varphi'(\theta,S)f_\theta(S)\\
&\geq \frac{3}{3+\epsilon}\sum_{\theta\in\Theta}\lambda(\theta)\sum_{S\subseteq[n]}\widetilde{\varphi}(\theta,S)f_\theta(S)\\
&= \frac{3}{3+\epsilon}\cdot Val(\widetilde{\varphi})\\
&\geq OPT-\frac{2\epsilon}{3},
\end{aligned}$$

where we used the upper bound $Val(\widetilde{\varphi})\leq 1$.

Existence of An ϵ-Optimal Solution of Small Support.

The above two steps of adjustment result in a feasible $\frac{2\epsilon}{3}$-optimal solution φ' to LP (5.7) that satisfies the following properties: (i) $x'_{\theta,i}=x^*_{\theta,i}=1$ whenever $u_i(\theta)\geq 0$; (ii) $x'_{\theta,i}=\frac{3}{3+\epsilon}\widetilde{x}_{\theta,i}=\frac{3}{3+\epsilon}x^*_{\theta,i}\geq\frac{\epsilon}{4n}$ when $x^*_{\theta,i}\geq\frac{\epsilon}{3n}$ and $\theta\in\Theta_i^-$; (iii) $x'_{\theta,i}=0$ when $x^*_{\theta,i}<\frac{\epsilon}{3n}$ and $\theta\in\Theta_i^-$. Utilizing such a φ' we show that there exists an ϵ-optimal solution $\overline{\varphi}$ to LP (5.7) such that the distribution $\overline{\varphi}_\theta$ is a K-uniform distribution for every θ, where $K=\frac{108n\log(2n|\Theta|)}{\epsilon^3}$.

Our proof is based on the probabilistic method. For each θ, independently take $K = \frac{108n\log(2n|\Theta|)}{\epsilon^3}$ samples from random variable $\varphi'(\theta)$, and let $\overline{\varphi}_\theta$ denote the corresponding empirical distribution. Obviously, $\overline{\varphi}_\theta$ is a K-uniform distribution. We claim that with strictly positive probability over the randomness of the samples, $\overline{\varphi}$ is feasible to LP (5.7) and achieves utility at least $Val(\varphi') - \frac{\epsilon}{3} \geq OPT - \epsilon$.

We first examine the objective value. Note that the objective value $Val(\varphi')$ can be viewed as the expectation of the random variable $\sum_{\theta\in\Theta}\lambda(\theta)f_\theta(S_\theta) \in [0,1]$, where S_θ follows the distribution of $\varphi'(\theta)$. Our sampling procedure generates K samples for the random variable $\{S_\theta\}_{\theta\in\Theta}$; therefore by the Hoeffding bound, with probability at least $1-\exp(-2K\epsilon^2/9) > 1-1/(2n|\Theta|)$, the empirical mean is at least $Val(\varphi') - \epsilon/3$.

Now we only need to show that all the persuasiveness constraints are preserved with high probability. First, if $x'_{\theta,i} = 0$, then $\overline{x}_{\theta,i}$ induced by $\overline{\varphi}$ also equals 0. This is because $x'_{\theta,i} = \mathbf{E}_{S\sim\varphi'(\theta)}\mathbb{I}(i \in S) = 0$ implies that i is not contained in any S from the support of $\varphi'(\theta)$, and therefore, also not contained in any sample. Similarly, $x'_{\theta,i} = 1$ implies $\overline{x}_{\theta,i} = 1$. To show that all the persuasiveness constraints hold, we only need to argue that $\overline{x}_{\theta,i} \leq x^*_{\theta,i}$ for every $\theta \in \Theta_i^-$ satisfying $x^*_{\theta,i} \geq \frac{\epsilon}{3n}$. This holds with high probability by tail bounds. In particular, $x'_{\theta,i} = \mathbf{E}_{S\sim\varphi'(\theta)}\mathbb{I}(i \in S)$ and we take K samples from $\varphi'(\theta)$. By the Chernoff bound, with probability at least

$$1-\exp(-\frac{K\epsilon^2 x'_{\theta,i}}{27}) \geq 1-\exp(-\frac{K\epsilon^3}{108n}) > 1-\frac{1}{2n|\Theta|},$$

the empirical mean $\overline{x}_{\theta,i}$ is at most $(1+\epsilon/3)x'_{\theta,i} = x^*_{\theta,i}$.

Note that there are at most $n|\Theta|$ choices of such θ, i. By the union bound, with probability at least $1-(n|\Theta|+1)/(2n|\Theta|) > 0$, $\overline{\varphi}$ satisfies all the persuasiveness constraints and thus is feasible for LP (5.7), and achieves objective value at least $Val(\varphi') - \frac{\epsilon}{3} \geq OPT - \epsilon$. So there must exist a feasible ϵ-optimal solution $\overline{\varphi}$ to LP (5.7) such that $\overline{\varphi}_\theta$ is K-uniform for every θ. This concludes our proof of Lemma 7.

5.3.5 The Sharp Contrast Between Private and Public Persuasion

A public signaling scheme π can be viewed as a special type of private signaling schemes in which each receiver must receive the same signal, i.e., only a public signal is sent. Overloading the notation of Section 5.3.3, we use Σ to denote the set of public signals and $\sigma \in \Sigma$ to denote a public signal. A public signaling scheme π is fully specified by $\{\pi(\theta,\sigma)\}_{\theta,\sigma}$, where $\pi(\theta,\sigma)$ denotes the probability of sending signal σ in state θ. Upon receiving a signal σ, each receiver performs the same Bayesian update and infers a posterior belief over the state of nature, as follows: the realized state is θ with probability $\lambda(\theta)\pi(\theta,\sigma)/\mathbf{Pr}(\sigma)$, where $\mathbf{Pr}(\sigma) = \sum_{\theta\in\Theta}\pi(\theta,\sigma)$. This induces a

subgame for each signal σ, one in which all receivers share the same belief regarding the state of nature.

Whereas in more general settings than ours, receivers may play a mixed Nash equilibrium in each subgame, our restriction to a setting with no externalities removes this complication. Given a posterior distribution on states of nature (say, one induced by a signal σ), our receivers face disjoint single-agent decision problems, each of which admits an optimal pure strategy. We assume that receivers break ties in favor of the sender (specifically, in favor of action 1), which results in a unique pure response for each receiver. Therefore, our solution concept here results in a unique action profile for each posterior distribution, and hence for each signal. A simple revelation-principle style argument then allows us to conclude that there is an optimal public signaling scheme which is *direct*, meaning that the public signals are action profiles, and *persuasive*, meaning that in the subgame induced by the signal $\sigma = (\sigma_1, \ldots, \sigma_n)$ each receiver i's optimal decision problem (which breaks ties in favor of action 1) solves to action σ_i.

Restricting attention to direct and persuasive public signaling schemes, each signal can also be viewed as a subset $S \subseteq [n]$ of receivers taking action 1. The sender's optimization problem can then be written as the following exponentially large linear program.

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta} \lambda(\theta) \sum_{S\subseteq[n]} \pi(\theta, S) f_\theta(S) & \\ \text{subject to} & \sum_{\theta\in\Theta} \lambda(\theta)\pi(\theta, S) \cdot u_i(\theta) \geq 0, & \text{for } S \subseteq [n] \text{ with } i \in S. \\ & \sum_{S\subseteq[n]} \pi(\theta, S) = 1, & \text{for } \theta \in \Theta. \\ & \pi(\theta, S) \geq 0, & \text{for } \theta \in \Theta; S \subseteq [n]. \end{array} \tag{5.13}$$

The first set of constraints are persuasiveness constraints corresponding to action 1. Like in LP (5.7), the persuasiveness constraints for action 0 are intentionally omitted from this LP. This omission is without loss when f_θ is non-decreasing for each state θ: if signal S with $i \notin S$ is such that receiver i prefers action 1 in the resulting subgame, then we can replace it with the signal $S \cup \{i\}$ without degrading the sender's utility. We remark that LP (5.13) and LP (5.7) only differ in their persuasiveness constraints.

We now consider the design of optimal public signaling schemes, and show a stark contrast with private signaling, both in terms of their efficacy at optimizing the sender's utility, and in terms of their computational complexity.

We start with an example illustrating how the restriction to public signaling can drastically reduce the sender's expected utility. The example is notably simple: two states of nature, and a binary sender utility function which is independent of the state of nature. We show a multiplicative gap of $\Omega(n)$, and an additive gap of $1 - \frac{1}{\Omega(n)}$, between the expected sender utility from the optimal private and public signaling schemes, where n is the number of receivers.

Example 2 (Inefficacy of Public Signaling Schemes)**.** *Consider an instance with n identical receivers and two states of nature $\Theta = \{\mathbf{H}, \mathbf{L}\}$. Each receiver has the same utility function, defined as follows: $u_i(\mathbf{H}) = 1$ and $u_i(\mathbf{L}) = -1$, for all i. The state of nature $\mathbf{H}$ occurs with probability $\frac{1}{n+1}$, and $\mathbf{L}$ occurs with probability $\frac{n}{n+1}$. The sender's utility function is $f_\theta(S) = f(S) = \min(|S|, 1)$. In other words, the sender gets utility 1 precisely when at least one receiver takes action 1.*

The persuasiveness constraints imply that each receiver can take action 1 with probability no more than $\frac{2}{n+1}$. This is achievable by always recommending action 1 to the receiver in state $\mathbf{H}$, and recommending action 0 with probability $\frac{1}{n}$ in state $\mathbf{L}$. The sender's expected utility depends on how these recommendations are correlated.

The optimal private scheme anti-correlates the receivers' recommendations in order to guarantee that at least one receiver takes action 1 always, which achieves an expected sender utility of 1, the maximum possible. Specifically, in state $\mathbf{H}$ the scheme always recommends action 1 to every receiver, and in state $\mathbf{L}$ the scheme chooses one receiver uniformly at random and recommends action 1 to that receiver, and action 0 to the other receivers.

We argue that no public scheme can achieve sender utility more than $\frac{2}{n+1}$. Indeed, since receivers are identical, our solution concept implies that they choose the same action for every realization of a public signal. Therefore, the best that a public scheme can do is to recommend action 1 to all receivers simultaneously with probability $\frac{2}{n+1}$ in aggregate, and recommend action 0 with the remaining probability, yielding an expected sender utility of $\frac{2}{n+1}$. This is achievable: in state $\mathbf{H}$ the scheme always recommends action 1 to every receiver, and in state $\mathbf{L}$ the scheme recommends action 1 to all receivers with probability $\frac{1}{n}$, and action 0 to all receivers with probability $1 - \frac{1}{n}$.

Our next result illustrates the computational barrier to obtaining the optimal public signaling scheme, even for additive sender utility functions. Our proof is inspired by a reduction in (Cheng et al., 2015) for proving the hardness of computing the best posterior distribution over Θ, a problem termed *mixture selection* in (Cheng et al., 2015), in a voting setting. That reduction is from the maximum independent set problem. Since a public signaling scheme is a combination of posterior distributions, one for each signal, we require a more involved reduction from a graph-coloring problem to prove our result.

Theorem 5.3.5. *Consider public signaling in our model, with sender utility function $f_\theta(S) = f(S) = \frac{|S|}{n}$. It is NP-hard to approximate the optimal sender utility to within any constant multiplicative factor. Moreover, there is no additive PTAS for evaluating the optimal sender utility, unless P = NP.*

Proof. We prove the theorem by reducing from the following NP-hard problem. (Khot & Saket, 2012) proves that for any positive integer k, any integer q such that $q \geq 2^k + 1$, and an arbitrarily small constant $\epsilon > 0$, given an undirected graph G, it is NP-hard to distinguish between the following two cases:

- **Case 1**: There is a q-colorable induced subgraph of G containing a $(1-\epsilon)$ fraction of all vertices, where each color class contains a $\frac{1-\epsilon}{q}$ fraction of all vertices.
- **Case 2**: Every independent set in G contains less than a $\frac{1}{q^{k+1}}$ fraction of all vertices.

Given a graph G with vertices $[n] = \{1, \ldots, n\}$ and edges E, we will construct a public persuasion instance so that the desired algorithm for approximating the optimal sender utility can be used to distinguish these two cases. Our construction is similar to that in (Cheng et al., 2015). We let there be n receivers, and let $\Theta = [n]$. In other words, both receivers and states of nature correspond to vertices of the graph. The prior distribution over states of nature is uniform — i.e., the realized state of nature is a uniformly-drawn vertex in the graph. We define the receiver utilities as follows: $u_i(\theta) = \frac{1}{2}$ if $i = \theta$; $u_i(\theta) = -1$ if $(i, \theta) \in E$; and $u_i(\theta) = -\frac{1}{4n}$ otherwise. We define the sender's utility function, with range $[0, 1]$, to be $f_\theta(S) = f(S) = \frac{|S|}{n}$. The following claim is proven in (Cheng et al., 2015).

Claim 3. *(Cheng et al., 2015) For any distribution $x \in \Delta_\Theta$, the set $S = \{i \in [n] : \sum_{\theta \in \Theta} x_\theta u_i(\theta) \geq 0\}$ is an independent set of G.*

Claim 3 implies that upon receiving any public signal with any posterior distribution x over Θ, the players who take action 1 always form an independent set of G. Therefore, if the graph G is from **Case 2**, the sender's expected utility in any public signaling scheme is at most $\frac{1}{q^{k+1}}$.

Now supposing that G is from **Case 1**, we fix the corresponding coloring of $(1-\epsilon)n$ vertices with colors $k = 1, \ldots, q$, and we use this coloring to construct a public scheme achieving expected sender utility at least $\frac{(1-\epsilon)^2}{q}$. The scheme uses $q+1$ signals, and is as follows: if θ has color k then deterministically send the signal k, and if θ is uncolored then deterministically send the signal 0. Given signal $k > 0$, the posterior distribution on states of nature is the uniform distribution over the vertices with color k — an independent set S_k of size $\frac{1-\epsilon}{q}n$. It is easy to verify that receivers $i \in S_k$ prefer action 1 to action 0, since $\sum_{\theta \in S_k} \frac{1}{|S_k|} u_i(\theta) = \frac{1}{|S_k|}(\frac{1}{2} - \frac{|S_k|-1}{4n}) > \frac{1}{4|S_k|} \geq 0$. Therefore, the sender's utility is $f(S_k) = \frac{|S_k|}{n} = \frac{1-\epsilon}{q}$ whenever $k > 0$. Since signal 0 has probability ϵ, we conclude that the sender's expected utility is at least $\frac{(1-\epsilon)^2}{q}$, as needed.

Since distinguishing **Case 1** and **Case 2** is NP-hard for arbitrarily large constants k and q, we conclude that it is NP-hard to approximate the optimal sender utility to within any constant factor. Moreover, by setting $k = 1, q = 3$, we conclude that the sender's utility cannot be approximated additively to within $(1-\epsilon)^2/3 - 1/3^2 > 1/9$, and thus there is no additive PTAS, unless P=NP.

□

Chapter 6

Persuasion in Security Games

Chapter 5 studies the algorithmic foundations for basic models of persuasion. In this chapter, we examine how these basic economic models can be applied to real-world security problems, particularly, the motivating domains described in Chapter 4. We will also illustrate how the specific domain features further complicate the problem and how we overcome these challenges by developing new algorithmic techniques.

6.1 Exploiting Informational Advantage to Deter Fare Evasion

In this section, we study how to improve a defender's utility by *strategically* revealing noisy information about each target's protection status to the attacker. We develop a two-stage security game model which abstracts the example described in Section 4.1. We then study when the defender can strictly benefit from such strategic signaling and how the defender can play both stages in a globally optimal fashion. Finally, we experimentally show that the two-state security game model allows the defender to achieve better utility than SSE in simulated random games.

6.1.1 A Two-Stage Security Game Model

Consider a security game where the defender allocates k security resources, possibly under scheduling constraints, to protect n targets. Players' strategies and the payoff structure are as described in Section 2.2.1. The game has two stages. The first stage is similar to regular security games, during which the defender commits to a mixed strategy. We now model the second stage — the signaling procedure. This stage can be viewed as a Bayesian persuasion game (Kamenica & Gentzkow, 2011), during which the defender persuades a rational attacker in order to yield a desired outcome. So we call it the *persuasion phase*. Specifically, for any $t \in [n]$ covered with probability x_t, let $Z = \{Z_c, Z_u\}$ be the set of events describing whether t is covered (Z_c) or not

(Z_u) and Σ be the set of all possible signals. A signaling scheme, with respect to target t, is a *randomized* map

$$f_c : Z \xrightarrow{rnd} \Sigma.$$

The set of probabilities

$$\{p(z, \sigma) : \ z \in Z, \ \sigma \in \Sigma\}$$

completely describes the map f, in which $p(z, \sigma)$ is the probability that event $z \in Z$ happens and signal $\sigma \in \Sigma$ is sent. Therefore, $\sum_\sigma p(z, \sigma) = \mathbb{P}(z)$, $\forall z \in Z$. Upon receiving a signal σ, the attacker infers a posterior distribution $\mathbb{P}(Z_c|\sigma) = \frac{p(Z_c,\sigma)}{p(Z_c,\sigma)+p(Z_u,\sigma)}$ and $\mathbb{P}(Z_u|\sigma) = \frac{p(Z_u,\sigma)}{p(Z_c,\sigma)+p(Z_u,\sigma)}$, and makes a decision among two actions: attack or not attack. For every target t, the defender seeks a signaling scheme w.r.t. t to maximize her expected utility on t.

Mathematically, a signal induces a posterior distribution on Z. Thus a signaling scheme can be viewed as a partition of the prior distribution $(x_t, 1 - x_t)$ into $|\Sigma|$ posteriors so that it maximizes the defender's utility on t. Like in Bayesian persuasion, we can w.l.o.g. focus on "direct" signaling schemes, as formalized in the following lemma.

Lemma 8. *(Kamenica & Gentzkow, 2011) There exists an optimal signaling scheme, w.r.t. any target t, that uses at most two signals, each resulting in an attacker best response of attacking and not attacking, respectively.*

As a result, a signaling scheme w.r.t. t can be characterized by

$$\begin{aligned} p(Z_c, \sigma_c) &= p \qquad & p(Z_c, \sigma_u) &= x_t - p; \\ p(Z_u, \sigma_c) &= q \qquad & p(Z_u, \sigma_u) &= 1 - x_t - q, \end{aligned}$$

in which $p \in [0, x_t], q \in [0, 1 - x_t]$ are variables. So the attacker infers the following expected utility: $\mathbb{E}(utility|\sigma_c) = \frac{1}{p+q}(pU_c^a + qU_u^a)$ and $\mathbb{E}(utility|\sigma_u) = \frac{1}{1-p-q}((x - p)U_c^a + (1 - x - q)U_u^a)$, where, for ease of notation, we drop the "t" in x_t and $U_{c/u}^{d/a}(t)$ when it is clear from context. W.l.o.g, let σ_c be a signal recommending the attacker to not attack, i.e., constraining $\mathbb{E}(utility|\sigma_c) \leq 0$, in which case both players get 0. Then the following LP parametrized by coverage probability x, denoted as $peLP_t(x)$ (Persuasion Linear Program), computes the optimal signaling scheme w.r.t. t:

$$\begin{aligned} \max \quad & (x - p)U_c^d + (1 - x - q)U_u^d \qquad (6.1) \\ s.t. \quad & pU_c^a + qU_u^a \leq 0 \\ & (x - p)U_c^a + (1 - x - q)U_u^a \geq 0 \\ & 0 \leq p \leq x \\ & 0 \leq q \leq 1 - x. \end{aligned}$$

This yields the attacker utility $\mathbb{P}(\sigma_u)\mathbb{E}(utility|\sigma_u) + \mathbb{P}(\sigma_c) \times 0 = (x-p)U_c^a + (1-x-q)U_u^a$ and defender utility $(x-p)U_c^d + (1-x-q)U_u^d$, w.r.t. t.

We propose the following two-stage Stackelberg security game model:

- Phase 1 (Scheduling Phase): the defender (randomly) schedules the resources by playing a mixed strategy $\boldsymbol{x} \in [0,1]^T$, and samples one pure strategy each round.
- Phase 2 (Persuasion Phase): $\forall t \in [n]$, the defender *commits* to an optimal signaling scheme w.r.t. t computed by $peLP_t(x_t)$ before the game starts, and then in each round, sends a signal on each target t according to the commitment.

During the play, the attacker first observes $\boldsymbol{x}$ by surveillance. Then he chooses a target t_0 to *approach* or *board* at some round, where the attacker receives a signal and decides whether to attack t_0 or not. Note that the model makes the following three assumptions. First, the defender is able to commit to a signaling scheme, and crucially will also follow the commitment. She is incentivized to do so because otherwise the attacker will not trust the signaling scheme, and thus may ignore signals. Then the game degenerates to a standard Stackelberg game. Second, the attacker breaks ties in favor of the defender. Similar to the definition of SSE, this is without loss of generality since if there is a tie among different choices, we can always make a tiny shift of the probability mass to make the choice preferred by the defender ϵ better than other choices. Third, we assume that the attacker cannot distinguish whether a target is protected or not when he approaches it.

With the persuasion phase, both of the defender and the attacker's payoff structures might be changed. Specifically, the defender's utility on any target t is the optimal objective *value* of the linear program $peLP_t(x)$, which is non-linear in x. Can the defender always *strictly* benefit by adding the persuasion phase? How can we compute the optimal mixed strategy in this new model? We answer these questions in the next two sections.

6.1.2 When Does Signaling Help?

In this section, fixing a marginal coverage x on a target t, we compare the defender's and attacker's utilities w.r.t. t in the following two different models:

- Model 1: the regular security game model, without persuasion (but the attacker can choose to not attack);
- Model 2: the two-stage security game model, in which the signaling scheme w.r.t. t is *optimal*.

The following notation will be used frequently in our comparisons and proofs (the index t is omitted when it is clear):

$$\begin{aligned}
\text{DefU}^{1/2}(t):&\ \text{defender's expected utility in Model 1/2;}\\
\text{AttU}^{1/2}(t):&\ \text{attacker's expected utility in Model 1/2;}\\
\text{U}^{\text{def/att}}(t):&\ = xU_c^{d/a} + (1-x)U_u^{d/a},\ \text{expected utility of}\\
&\ \text{defense/attack, } \textit{if attacker attacks } t.
\end{aligned}$$

Note that $\text{AttU}^1 = \max(\text{U}^{\text{att}}, 0)$ may not equal U^{att} since the attacker chooses to not attack if $\text{U}^{\text{att}} < 0$. Similarly, DefU^1 may not equal to U^{def}.

Defender's Utility

First, we observe that the defender will never be worse off in Model 2 than Model 1 w.r.t. t.

Proposition 1. *For any* $t \in [n]$, $\text{DefU}^2 \geq \text{DefU}^1$.

Proof. If $\text{U}^{\text{att}} \geq 0$, then $p, q = 0$ is a feasible solution to $peLP_t(x)$ in formula (6.1), which achieves a defender utility $xU_c^d + (1-x)U_u^d = \text{DefU}^1$. So $\text{DefU}^2 \geq \text{DefU}^1$.

If $\text{U}^{\text{att}} < 0$, the attacker will choose to not attack in Model 1, so $\text{DefU}^1 = 0$. In this case, $p = x, q = 1 - x$ is a feasible solution to $peLP_t(x)$, which achieves a defender utility 0. So $\text{DefU}^2 \geq 0 = \text{DefU}^1$. □

However, the question is whether the defender will always *strictly* benefit w.r.t. t from the persuasion phase. The following theorem gives a succinct characterization.

Theorem 6.1.1. *For any* $t \in [n]$ *with marginal coverage* $x \in [0, 1]$, $\text{DefU}^2 > \text{DefU}^1$*, if and only if:*

$$U^{att}(U_c^d U_u^a - U_c^a U_u^d) < 0. \tag{6.2}$$

Proof. The inequality Condition (6.2) corresponds to the following four cases:

1. $\text{U}^{\text{att}} > 0, U_u^d \geq 0, U_c^d U_u^a - U_c^a U_u^d < 0$;
2. $\text{U}^{\text{att}} > 0, U_u^d < 0, U_c^d U_u^a - U_c^a U_u^d < 0$;
3. $\text{U}^{\text{att}} < 0, U_u^d \geq 0, U_c^d U_u^a - U_c^a U_u^d > 0$;
4. $\text{U}^{\text{att}} < 0, U_u^d < 0, U_c^d U_u^a - U_c^a U_u^d > 0$.

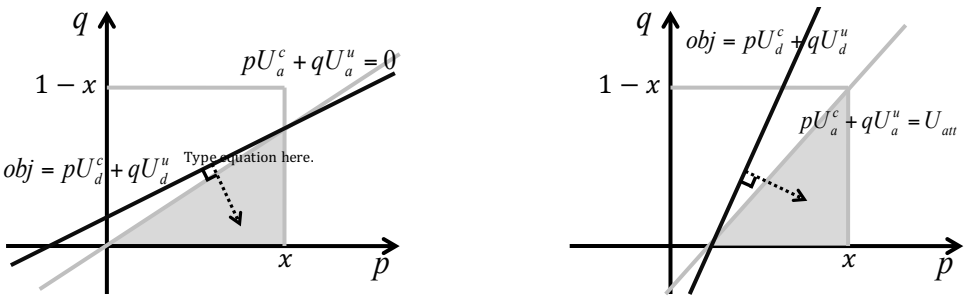


Figure 6.1: Feasible regions (gray areas) and an objective function gaining strictly better defender utility than SSE for the case $\mathrm{U}^{\mathrm{att}} > 0$ (Left) and $\mathrm{U}^{\mathrm{att}} < 0$ (Right).

Case 1 obviously does not happen, since $U_c^d U_u^a - U_c^a U_u^d > 0$ when $U_c^d > U_u^d \geq 0$ and $U_u^a > 0 > U_c^a$. Interestingly, cases 2–4 correspond exactly to all the three possible conditions that make $\mathrm{DefU}^2 > \mathrm{DefU}^1$. We now give a geometric proof. Instead of $peLP_t(x)$, we consider the following equivalent LP:

$$\begin{aligned} \min \quad & pU_c^d + qU_u^d \\ s.t. \quad & pU_c^a + qU_u^a \leq 0 \\ & pU_c^a + qU_u^a \leq \mathrm{U}^{\mathrm{att}} \\ & 0 \leq p \leq x \\ & 0 \leq q \leq 1 - x, \end{aligned}$$

so that $\mathrm{DefU}^2 = \mathrm{U}^{\mathrm{def}} - \mathrm{Opt}$. Figure 6.1 plots the feasible region for the cases $\mathrm{U}^{\mathrm{att}} > 0$ and $\mathrm{U}^{\mathrm{att}} < 0$, respectively. Note that the vertex $(x, 0)$ can never be an optimal solution in either case, since the feasible point $(x - \epsilon, \epsilon)$ for tiny enough $\epsilon > 0$ always achieves strictly smaller objective value, assuming $U_c^d > U_u^d$. When $\mathrm{U}^{\mathrm{att}} > 0$, the attacker chooses to attack, resulting in $\mathrm{DefU}^1 = \mathrm{U}^{\mathrm{def}}$. So to strictly increase the defender's utility is equivalent to making $\mathrm{Opt} < 0$ for the above LP. That is, we only need to guarantee that the optimal solution is *not* the origin $(0, 0)$ (a vertex of the feasible polytope). This happens when $U_u^d < 0$, and the slope of $obj = pU_c^d + qU_u^d$ is less than the slope of $0 = pU_c^a + qU_u^a$, that is, $U_c^d/U_u^d - U_c^a/U_u^a > 0$. These conditions correspond to the case 2. In this case, the defender gains *extra* utility $-\mathrm{Opt} = -\frac{z}{U_u^a}(U_u^a U_c^d - U_c^a U_u^d) > 0$ by adding the persuasion phase.

When $\mathrm{U}^{\mathrm{att}} < 0$, the attacker chooses to not attack, resulting in $\mathrm{DefU}^1 = 0$. To increase the defender's utility, we have to guarantee $\mathrm{Opt} < \mathrm{U}^{\mathrm{def}}$. Note that the vertex $(x, 1 - x)$ yields exactly an objective $\mathrm{U}^{\mathrm{def}}$, so we only need to guarantee the optimal solution is the vertex $(\frac{\mathrm{U}^{\mathrm{att}}}{U_c^a}, 0)$. This happens either when $U_u^d \geq 0$ (corresponding to case 3 in which case $U_c^d U_u^a - U_c^a U_u^d > 0$ holds naturally) or when $U_u^d < 0$ and the slope of $obj = pU_c^d + qU_u^d$ is greater than the slope

of $0 = pU_c^a + qU_u^a$. That is, $-U_c^d/U_u^d > -U_c^a/U_u^a$. This corresponds to case 4 above. In such cases, the defender gains *extra* utility $\text{U}^{\text{def}} - \text{Opt} = -\frac{1-x}{U_c^a}(U_u^a U_c^d - U_c^a U_u^d) > 0$ by adding the persuasion phase.

When $\text{U}^{\text{att}} = 0$, the possible optimal vertices are $(0, 0)$ and $(x, 1-x)$, which corresponds to the defender utility 0 and U^{def}, respectively. So $\text{DefU}^2 = \max\{0, \text{U}^{\text{def}}\}$ at optimality, which equals to DefU^1 assuming the attacker breaks ties in favor of the defender. □

Interpreting the Condition in Theorem 6.1.1

Inequality (6.2) immediately yields that the defender does not benefit from persuasion in zero-sum security games, since $U_c^d U_u^a - U_c^a U_u^d = 0$ for any target in zero-sum games. Intuitively, this is because there are no posterior distributions, and thus signals, where the defender and attacker can cooperate due to the strictly competitive nature of zero-sum games.

One case of the Inequality (6.2) is $\text{U}^{\text{att}} > 0$ and $U_c^d U_u^a - U_c^a U_u^d < 0$. To interpret the latter, let us start from a zero-sum game, which assumes $-U_u^d = U_u^a > 0$ and $U_c^d = -U_c^a > 0$. Then the condition $U_c^d U_u^a - U_c^a U_u^d = U_c^d U_u^a - (-U_c^a)(-U_u^d) < 0$ could be achieved by making $-U_u^d > U_u^a$ or $U_c^d < -U_c^a$. That is, the defender values a target more than the attacker ($-U_u^d > U_u^a$), e.g., the damage to a flight causes more utility loss to the defender than the utility gained by the attacker, or the defender values catching the attacker less than the cost to the attacker ($U_c^d < -U_c^a$), e.g., the defender does not gain much benefit by placing a violator in jail but the violator loses a lot. In such games, if the attacker has incentives to attack (i.e., $\text{U}^{\text{att}} > 0$), the defender can "persuade" him to not attack.

Another case of Condition 2 is $\text{U}^{\text{att}} < 0$ and $U_c^d U_u^a - U_c^a U_u^d > 0$. In contrast to the situation above, this is when the defender values a target less than the attacker (e.g., a fake target or honey pot) but cares more about catching the attacker. Interestingly, the defender benefits when the attacker does not want to attack (i.e., $\text{U}^{\text{att}} < 0$), but the defender "entices" him to commit an attack in order to catch him.

Attacker's Utility

Now we compare the attacker's utilities w.r.t. t in Model 1 and Model 2. Recall that Proposition 1 shows the defender will never be worse off. A natural question is: whether the attacker can be strictly better off? The attacker will never be worse off under *any signaling scheme*. Intuitively, this is because he could just ignore any signals. Mathematically, this holds simply by observing the constraints in $peLP_t(x)$ Formulation 6.1:

1. when $\text{U}^{\text{att}} \geq 0$, $\text{AttU}^1 = \text{U}^{\text{att}} = xU_c^a + (1-x)U_u^a$ and $\text{AttU}^2 = (x-p)U_c^a + (1-x-q)U_u^a$, so $\text{AttU}^1 - \text{AttU}^2 = pU_c^a + qU_u^a \leq 0$;

2. when $\text{U}^{\text{att}} < 0$, $\text{AttU}^2 = (x-p)U_c^a + (1-x-q)U_u^a \geq 0 = \text{AttU}^1$.

Note that the above conclusion holds without requiring the signaling scheme to be optimal, since the derivation only uses feasibility constraints. Interestingly, if the defender does persuade optimally, then equality holds.

Theorem 6.1.2. *Given any target $t \in [n]$ with marginal coverage $x \in [0,1]$, we have* $\text{AttU}^1 = \text{AttU}^2 = \max(0, \text{U}^{\text{att}})$.

Proof. From $peLP_t(x)$ we know that $\text{AttU}^2 = \text{U}^{\text{att}} - (pU_c^a + qU_u^a)$. The proof is divided into three cases. When $\text{U}^{\text{att}} > 0$ (left panel in Figure 6.1), we have $\text{AttU}^1 = \text{U}^{\text{att}}$. As argued in the proof of Theorem 6.1.1, the optimal solution can never be the vertex $(x, 0)$. So the only possible optimal vertices are $(0,0)$ and $(x, -x\frac{U_c^a}{U_u^a})$, both of which satisfy $pU_c^a + qU_u^a = 0$. So $\text{AttU}^2 = \text{U}^{\text{att}} - (pU_c^a + qU_u^a) = \text{U}^{\text{att}} = \text{DefU}^1$. When $\text{U}^{\text{att}} < 0$ (right panel in Figure 6.1),we have $\text{AttU}^1 = 0$. The only possible optimal vertices are $(x, 1-x)$ or $(-\frac{\text{U}^{\text{att}}}{U_c^a}, 0)$, both of which satisfies $pU_c^a + qU_u^a = \text{U}^{\text{att}}$. So $\text{AttU}^2 = 0 = \text{AttU}^1$. For the case $\text{U}^{\text{att}} = 0$, a similar argument holds. To sum up, we always have $\text{AttU}^1 = \text{AttU}^2$. □

6.1.3 Computing the Optimal Defender Strategy

As we have seen so far, the defender can strictly benefit from persuasion in the two-stage security game model. Here comes the natural question for computer scientists: how can we compute the optimal mixed strategy? We answer the question in this section, starting with an example showing that the defender's optimal mixed strategy in the two-stage model is *different* from the SSE in its standard security game model.

Example 3. *Consider a security game with payoff matrix in Table 6.1.*

	U_c^d	U_u^d	U_c^a	U_u^a
t_1	1	-2	-1	1
t_2	3	-5	-3	5
t_3	1	-4	-2	4
t_4	0	-0.5	-2	1

Table 6.1: Payoff table for the constructed game

Assume that there are two resources, and the feasible pure strategies are $A_1 = (t_1, t_2)$, $A_2 = (t_2, t_3)$ and $A_3 = (t_3, t_4)$. Let $\boldsymbol{p} = (p_1, p_2, p_3)$ denote a mixed strategy where p_i is the probability of taking action A_i. After simple calculations, one can compute the Strong Stackelberg Equilibrium (SSE) as $\boldsymbol{p} = (\frac{3}{8}, \frac{7}{32}, \frac{13}{32})$ with coverage probability vector $\boldsymbol{x} = (\frac{3}{8}, \frac{19}{32}, \frac{5}{8}, \frac{13}{32})$. The

attacker's utility is $(\frac{1}{4}, \frac{1}{4}, \frac{1}{4}, -\frac{7}{32})$ *and the defender's utility is* $(-\frac{7}{8}, -\frac{1}{4}, -\frac{7}{8}, -\frac{19}{64})$*, so the attacker will attack* t_2*.*

Now, if we add the persuasion phase as in Model 2, the optimal mixed strategy is $\boldsymbol{p} = (\frac{3}{8}, \frac{3}{8}, \frac{1}{4})$ *with coverage probability vector* $\boldsymbol{x} = (\frac{3}{8}, \frac{3}{4}, \frac{5}{8}, \frac{1}{4})$*. The attacker's utility is* $(\frac{1}{4}, -1, \frac{1}{4}, \frac{1}{4})$ *and defender's utility is* $(-\frac{1}{2}, 1, -\frac{1}{4}, -\frac{1}{8})$*, so the attacker will attack* t_4*, in favor of the defender's preference. So the defender's utility changes from* $-\frac{1}{4}$ *in Model 1 to* $-\frac{1}{8}$ *in Model 2.*

Therefore, we define the following solution concept.

Definition 3. *The optimal defender mixed strategy and signaling scheme in the two-stage Stackelberg security game, together with the attacker's best response, form an equilibrium called the* Strong Stackelberg Equilibrium with Persuasion (peSSE).

Proposition 1 yields that, by adding the persuasion phase, the defender's utility will not be worse off under any mixed strategy, and specifically under the SSE mixed strategy. This yields the following performance guarantee of peSSE.

Proposition 2. *Given any security game, the defender's utility in peSSE is at least the defender's utility in SSE.*

Now we consider the computation of peSSE. Note that the optimal signaling scheme can be computed by LP 6.1 for any target t with given coverage probability x_t. The main challenge is how to compute the optimal mixed strategy in Phase 1. Assume that the defender's (leader) mixed strategy, represented as a marginal coverage vector over target set $[n]$, lies in a polytope $\mathcal{P}_d$. [1] With a bit of abuse of notation, let us use $peLP_t(x_t)$ to denote also the optimal objective value of the persuasion LP, as a function of x_t. Let

$$\mathrm{U}^{\mathrm{att}}(t, x) = xU_c^a(t) + (1 - x)U_u^a(t)$$

be the attacker's expected utility, if he attacks, as a *linear* function of x.

Recall that, given a mixed strategy $\boldsymbol{x} \in [0, 1]^T$, the defender's utility w.r.t. t is $peLP_t(x_t)$ and the attacker's utility w.r.t. t is $\max(\mathrm{U}^{\mathrm{att}}(t, x_t), 0)$ (Theorem 6.1.2). Similar to the framework in

[1] Note that a polytope can always be represented by linear constraints (though possibly exponentially many). For example, a simple case is the games in which pure strategies are arbitrary subsets $A \subseteq [n]$ with cardinality $|A| \leq k$, $\mathcal{P}_d$ can be represented by $2T + 1$ linear inequalities: $\sum_i x_i \leq k$ and $\mathbf{0} \leq \boldsymbol{x} \leq \mathbf{1}$. However, $\mathcal{P}_d$ can be complicated in security games, such that it is NP-hard to optimize a linear objective over $\mathcal{P}_d$ (Xu, 2016). Finding succinct representations of $\mathcal{P}_d$ plays a key role in the computation of SSE.

(Conitzer & Sandholm, 2006), we define the following optimization problem for every target t, denoted as OPT_t:

$$\begin{aligned} \max \quad & peLP_t(x_t) \\ s.t. \quad & \max(\mathrm{U}^{\mathrm{att}}(t, x_t), 0) \geq \max(\mathrm{U}^{\mathrm{att}}(t', x_{t'}), 0) \, \forall t' \\ & \boldsymbol{x} \in \mathcal{P}_d, \end{aligned} \tag{6.3}$$

which computes a defender mixed strategy maximizing the defender's utility on t, subject to: 1. the mixed strategy is achievable; 2. attacking t is the attacker's best response. Notice that some of these optimization problems may be infeasible. Nevertheless, at least one of them is feasible. The peSSE is obtained by solving these T optimization problems and picking the best solution among those OPT_t's.

To solve optimization problem (6.3), we have to deal with non-linear constraints and the specific objective $peLP_t(x_t)$, which is the optimal objective value of another LP. We first simplify the constraints to make them linear. In particular, the following constraints

$$\max(\mathrm{U}^{\mathrm{att}}(t, x_t), 0) \geq \max(\mathrm{U}^{\mathrm{att}}(t', x_{t'}), 0), \forall t' \in [n]$$

can be split into two cases, corresponding to $\mathrm{U}^{\mathrm{att}}(t, x_t) \geq 0$ and $\mathrm{U}^{\mathrm{att}}(t, x_t) \leq 0$ respectively, as follows,

CASE 1	CASE 2
$\mathrm{U}^{\mathrm{att}}(t, x_t) \geq 0$ $\mathrm{U}^{\mathrm{att}}(t, x_t) \geq \mathrm{U}^{\mathrm{att}}(t', x_{t'}), \forall t'$	$\mathrm{U}^{\mathrm{att}}(t', x_{t'}) \leq 0, \forall t'$

Now, the only problem is to deal with the objective function in Formulation (6.3). Here comes the crux.

Lemma 9. *For any $t \in [n]$, $peLP_t(x)$ is increasing in x for any $x \in (0, 1)$.*

Proof. For ease of notation, let $f(x) = peLP_t(x)$. We show that for any sufficiently small $\epsilon > 0$ (so that $x + \epsilon < 1$), $f(x + \epsilon) \geq f(x)$. Fixing x, if the optimal solution for $peLP_t(x)$, say p^*, q^*, satisfies $q^* = 0$, then we observe that p^*, q^* is also feasible for $peLP_t(x+\epsilon)$. As a result, plugging p^*, q^* in $peLP_t(x+\epsilon)$, we have $f(x+\epsilon) \geq (x-p^*)U_c^d + (1-x-q^*)U_u^d + \epsilon(U_c^d - U_c^d) \geq f(x)$ since $\epsilon(U_c^d - U_c^d) \geq 0$. On the other hand, if $q^* > 0$, then for any small $\epsilon > 0$ (specifically, $\epsilon < q^*$), $p^* + \epsilon, q^* - \epsilon$ is feasible for $peLP_t(x + \epsilon)$. Here the only need is to check the feasibility constraint $(p^* + \epsilon)U_c^a + (q^* - \epsilon)U_u^a = p^*U_c^a + q^*U_u^a + \epsilon(U_c^a - U_u^a) \leq 0$, which holds since

$\epsilon(U_c^a - U_u^a) \leq 0$. This feasible solution achieves an objective value equaling $f(x)$. Therefore, we must have $f(x+\epsilon) \geq f(x)$. □

The intuition behind Lemma 9 is straightforward — the defender should always get more utility by protecting a target more. However, this actually does not hold in standard security games. Simply consider a target with $U_c^d = 2$, $U_u^d = -1$ and $U_c^a = -1$, $U_u^a = 1$. If the target is covered with probability 0.4, then in expectation both the attacker and defender get 0.2; however, if the target is covered with probability 0.6, the attacker will not attack and both of them get 0. Therefore, the monotonicity in Lemma 9 is really due to the signaling scheme.

Back to the optimization problem (6.3), here comes our last key observation: the monotonicity property in Lemma 9 reduces the problem to an LP. Specifically, the following lemma is a simple consequence of the monotonicity.

Lemma 10. *Maximizing the* increasing *function $peLP_t(x_t)$ over any feasible region $\mathcal{D}$ reduces to directly maximizing x_t over $\mathcal{D}$ and then plugging in the optimal x_t to $peLP_t(x_t)$.*

To this end, we summarize the main results in this section. The following theorem essentially shows that computing peSSE efficiently reduces to computing SSE (see (Conitzer & Sandholm, 2006) for a standard way to compute SSE by multiple LPs). In other words, adding the persuasion phase does not increase the computational complexity.

Theorem 6.1.3. *For any security game, the* Strong Stackelberg Equilibrium with Persuasion (peSSE)*, defined in Definition 3, can be computed by multiple LPs.*

Proof. According to Lemma 9 and 10, Algorithm 4, based on multiple LPs, computes the peSSE. □

6.1.4 Experiments

In this section, we compare SSE and peSSE on randomly generated security games. Our simulations aim to compare the two concepts, SSE and peSSE, in games with various payoff structures.

To generate payoffs, we follow most security game papers and use the covariance random payoff generator (Nudelman, Wortman, Shoham, & Kevin, 2004), but with a slight modification. Let $\mu[a,b]$ denote a uniform distribution on interval $[a,b]$. Then we randomly generate the following random payoffs: $U_c^d \sim \mu[0,r]$, $U_u^d \sim \mu[-10,0]$, $U_c^a = aU_c^d \times \frac{10}{r} + b\mu[-10,0]$ (set $U_c^d \times \frac{10}{r} = 0$ if $r = 0$) and $U_u^a = aU_u^d + b\mu[0,10]$, where $a = cov, b = \sqrt{1-a^2}$. Here $cov \in [-1,0]$ is the covariance parameter between the defender's reward (or penalty) and the attacker's penalty (or reward). So $cov = 0$ means completely uncorrelated payoffs while $cov = -1$ and $r = 10$ means a zero-sum game. By setting $U_c^d \in [0,r]$ while $U_c^a \in [0,10]$, we intentionally

Algorithm 4: Computing peSSE

1: For every target $t \in [n]$, compute the optimal objectives for the following two LPs:

$$\begin{aligned} \max \quad & x_t \\ s.t. \quad & \mathrm{U}^{\mathrm{att}}(t, x_t) \geq 0 \\ & \mathrm{U}^{\mathrm{att}}(t, x_t) \geq \mathrm{U}^{\mathrm{att}}(t', x_{t'}), \forall t' \in [n] \\ & \boldsymbol{x} \in \mathcal{P}_d \end{aligned} \tag{6.4}$$

and

$$\begin{aligned} \max \quad & x_t \\ s.t. \quad & \mathrm{U}^{\mathrm{att}}(t', x_{t'}) \leq 0, \forall t' \in [n] \\ & \boldsymbol{x} \in \mathcal{P}_d. \end{aligned} \tag{6.5}$$

Let $x^*_{t,1}$, $x^*_{t,2}$ be the optimal objective value for LP (6.4), LP (6.5) respectively. $x^*_{t,i} = null$ if the corresponding LP is infeasible.

2: Choose the non-null $x^*_{t,i}$, denoted as x^*, that maximizes $peLP_t(x^*_{t,i})$ over $t \in [n]$ and $i = 1, 2$. The optimal mixed strategy that achieves x^* in one of the above LPs is the peSSE mixed strategy.

capture the defender's "overall" value of catching the attacker by the parameter r. Standard covariance payoff (Nudelman et al., 2004) fixes $r = 10$, but Theorem 1 suggests that r may affect the utility difference between SSE and peSSE.

In all the simulations, every game has 8 targets and 3 resources, and the attacker has the option to not attack. We simulate two different kinds of pure strategies, which results in two types of games:

1. **Uni**form Strategy **G**ame (**UniG**): in such games, a pure strategy is any subset of targets with cardinality at most 3.

2. **Ran**dom Strategy **G**ame (**RanG**): for each game we randomly generate 6 pure strategies, each of which is a subset of targets with cardinality at most 3. Each target is guaranteed to be covered by at least one pure strategy.

We set $r = 0, 1, \ldots, 10$ and $cov = 0, -0.1, -0.2, \ldots, -1$. For each parameter instance, i.e., r and cov, *100* random security games are simulated. As a result, in total $2 \times 100 \times 11^2 = 24,200$ (2 types of games, 11^2 parameter combinations and 100 games per case) random security games are tested in our experiments. We find that the UniG and RanG games have similar experimental performance, except that RanG games have a lower utility at a given parameter instance. This

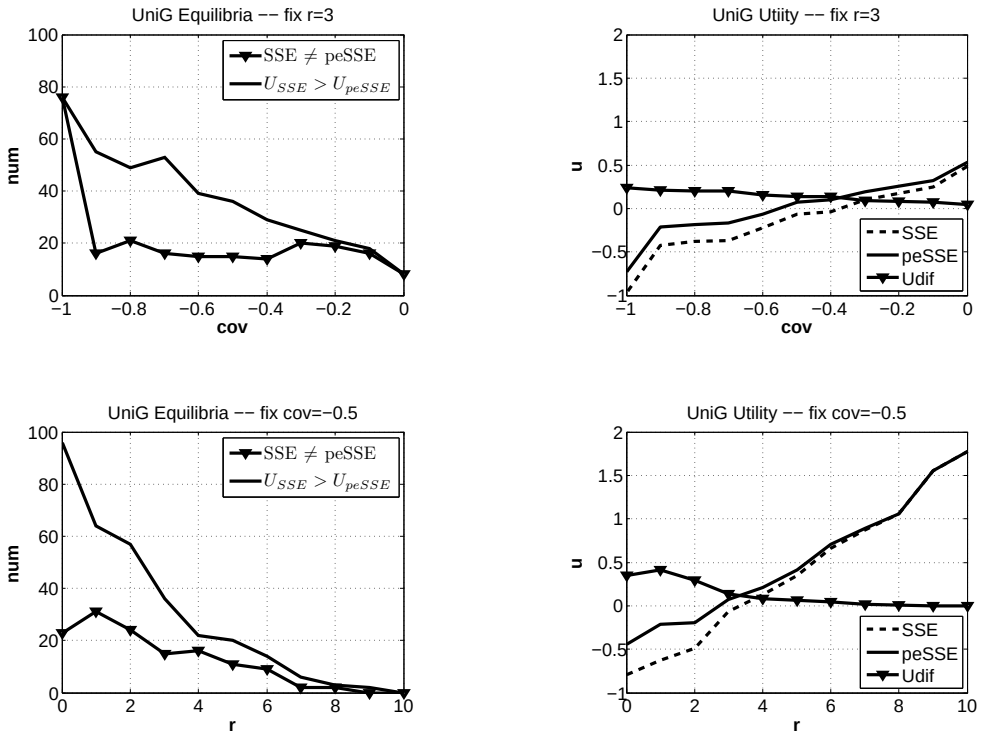


Figure 6.2: Comparison between SSE and peSSE: fixed parameter $r = 3$ (upper) and fixed parameter $cov = -0.5$. The trend is similar for different r or cov, except the utility scales are different.

is reasonable since UniG games are relaxations of the RanG games in terms of the set of pure strategies. So we only show results for UniG to avoid repetition.

Figure 6.2 gives a comprehensive comparison of the difference between SSE and peSSE. All these performances are averaged over 100 games. These figures suggest the following empirical conclusions as expected (note that the trends reflected in the figures are basically similar for different r or cov, except the utility scales are different):

- In the left two panels, the line $SSE \neq peSSE$ describes the number of games within 100 simulations that have different SSE and peSSE mixed strategies. This number seems not very sensitive to the parameter cov (note that games with $cov = -1$ are not zero-sum when $r = 3$), but increases as r decreases. That is, when the defender cares less about catching the attacker, then persuading the attacker to not attack benefits the defender more.

- The line $U_{SSE} > U_{peSSE}$ in the left two panels describes how many games have *strictly* greater peSSE utility than SSE utility. This number increases as cov or r decreases. That is, if the defender cares less about catching the attacker or the game becomes more competitive

(i.e., cov decreases), then the defender benefits more from strategic signaling. Note that the *Udif* lines in the right two panels also show the same trend.

- The right two panels show that persuasion usually helps more when the defender's SSE utility is less. Specifically, peSSE can increase the SSE utility by about half when r is small with fixed $cov = -0.5$ (right-lower panel).

6.2 Exploiting Informational Advantages to Combat Poaching

In this section, we study how to improve a defender's utility via strategic signaling in a different setting, motivated by the emerging application of utilizing mobile sensors for patrolling (the example in Section 4.2). This setting differs from that of the previous section in the following key aspects. First, here we assume that the attacker (e.g., a poacher) can observe whether a patroller is at a target or not before he attacks the target (e.g., whether a ranger is patrolling the area or not); while in the previous section, the attacker cannot observe whether a target is protected or not before he attacks. Second, here the defender has a limited number of signaling devices (e.g., mobile sensors) and we have to optimally place these signaling devices at targets; while previously, the defender can have every target signal noisy information about its own protection status.

These differences necessitate a different game model with strategic signaling. In particular, we propose the *Sensor-Empowered security Game* (SEG) model in this section. SEG captures the *joint* allocation of human patrollers and mobile sensors, and abstracts the example described in Section 4.2. Sensors differ from patrollers in that they cannot directly interdict attacks, but they can notify nearby patrollers about the attack (if any) and strategically signal to the attacker in order to deter attacks. On the technical side, we first illustrate the challenges in solving the new SEG model by proving its NP-hardness even for zero-sum cases. We then develop a scalable algorithm `SEGer` based on the branch-and-price framework with two key novelties: (1) a novel MILP formulation for the slave; (2) an efficient relaxation of the problem for pruning. To further accelerate `SEGer`, we design a *faster* combinatorial algorithm for the slave problem, which is provably a constant-approximation to the slave problem in zero-sum cases and serves as a useful heuristic for general-sum SEGs. We experimentally demonstrate the benefit of utilizing mobile sensors via simulations.

6.2.1 The Model

Basic Setup. Consider a security game played between a defender (she) and an attacker (he). The defender possesses k human *patrollers* and m mobile *sensors*. She aims to protect n targets,

whose underlying geographic structure is captured by an undirected graph G. We use $[n]$ to denote the set of all targets, i.e., all vertices. The attacker seeks to attack one target. Let $U^{d/a}_{+/-}(i)$ denote the defender/attacker (d/a) payoff when the defender successfully protects/fails to protect $(+/-)$ the attacked target i.[2] Assume $U^d_+(i) \geq 0 > U^d_-(i)$ and $U^a_+(i) \leq 0 < U^a_-(i)$ for every i. Sensors *cannot* directly interdict an attack; however, they can inform patrollers to come when detecting the attacker at a target. Let integer $\tau > 0$ be the *intervention distance* such that a sensor-informed patroller within distance τ to the attacked target can successfully come to intervene in the attack. If there is no patroller within distance τ to the attacked target, the target is not protected despite being covered by a sensor. So a target covered by some resource (i.e., sensors) is not necessarily protected, which is a key difference between SEGs and classical security games. We assume that sensors are visible. Therefore, the attacker knows whether a target is covered by a sensor or not, upon visiting the target.

Defender's Action Space of Resource Allocation. We assume that any patroller or sensor can be assigned to cover any target on G without scheduling restrictions. Therefore, a *defender pure strategy* covers an arbitrary subset of k vertices with patrollers and another subset of m vertices with sensors. For convenience, we call both patrollers and sensors *resources*. W.l.o.g., we assume that the defender never places more than one resource at any target (otherwise, reallocating one resource to any uncovered target would only do better). Targets in SEGs have 4 possible states: (1) covered by a patroller (state θ_+); (2) uncovered by any resource (state θ_-); (3) covered by a sensor and at least one patroller is within distance τ (state θ_{s+}); (4) covered by a sensor but no patroller is within distance τ (state θ_{s-}). Note that only state θ_+, θ_{s+} mean successful defense. Let $\Theta = \{\theta_+, \theta_-, \theta_{s+}, \theta_{s-}\}$ denote the set of all states. Any resource allocation uniquely determines the state for each target and vice versa. Therefore we can equivalently use a state vector $\mathbf{e} \in \Theta^n$ to denote a defender pure strategy. Let $e_i \in \Theta$ denote the state of target $i \in [n]$ and $\mathcal{E} \subseteq \Theta^n$ denote the set of defender pure strategies. A *defender mixed strategy* is a distribution over the exponentially large set $\mathcal{E}$.

Mobile Sensor Signaling. SEGs naturally integrate the sensor functionality of strategic signaling, which can be easily implemented for many types of mobile sensors (e.g., UAVs). Let Σ denote the set of possible signals that a sensor could send (e.g, noise, warning lights, etc.). Let $\Theta_s = \{\theta_{s+}, \theta_{s-}\}$ denote the set of possible states when a sensor covers the target. A *signaling scheme*, w.r.t. target i, is a randomized map

$$\pi_i : \Theta_s \xrightarrow{\text{rnd}} \Sigma,$$

[2]The utility notation $U^{d/a}_{+/-}(i)$ is different from the standard notation $U^{d/a}_{c/u}(i)$ of classic security games as described in Section 2.2.1. This is to avoid confusion because in SEGs, successfully protecting a target is not the same as covering the target with a security guard. For example, if a target is covered by a UAV and meanwhile a security guard is nearby, the target is also successfully protected.

which is characterized by variables $\{\pi_i(e_i, \sigma_i)\}_{e_i \in \Theta_s, \sigma_i \in \Sigma}$. Here $\pi_i(e_i, \sigma_i)$ is the joint probability that target i is in state $e_i \in \Theta_s$ and signal $\sigma_i \in \Sigma$ is sent. So $\sum_{\sigma_i \in \Sigma} \pi_i(e_i, \sigma_i)$ must equal $\mathbb{P}(e_i)$, the marginal probability that target i is in state e_i. A sensor at target i first determines its state $e_i \in \Theta_s$ and then sends a signal σ_i with probability $\pi_i(e_i, \sigma_i)/\mathbb{P}(e_i)$. We assume that the defender *commits* to a signaling scheme and the rational attacker is aware of the commitment.

Upon observing signal σ_i, the attacker updates his belief on the target state: $\mathbb{P}(\theta_{s+}|\sigma_i) = \frac{\pi_i(\theta_{s+}, \sigma_i)}{\pi_i(\theta_{s+}, \sigma_i) + \pi_i(\theta_{s-}, \sigma_i)}$ and $\mathbb{P}(\theta_{s-}|\sigma_i) = 1 - \mathbb{P}(\theta_{s+}|\sigma_i)$, and derives expected utility

$$\text{AttU}(\sigma_i) = U^a_+(i) \cdot \mathbb{P}(\theta_{s+}|\sigma_i) + U^a_-(i) \cdot \mathbb{P}(\theta_{s-}|\sigma_i).$$

The attacker will attack target i if $\text{AttU}(\sigma_i) > 0$. When $\text{AttU}(\sigma_i) < 0$, the rational attacker chooses to not attack, in which case both players get utility 0. We assume that the attacker breaks tie in favor of the defender when $\text{AttU}(\sigma_i) = 0$. This is without loss of generality because the defender can always slightly tune the probabilities to favor her preferred attacker action.

As illustrated in Lemma 8 of the previous section, there always exists an optimal signaling scheme (w.r.t. a target) that uses at most two signals, each resulting in an attacker best response of attacking and not attacking, respectively. In our previous example of Section 6.2.3.1, an *alert* signal results in not attacking while a *quiet* signal result in attacking.

Attacker's Action Space. We assume that the defender *commits* to a mixed strategy (i.e., randomized resource allocation) and signaling schemes. The attacker is aware of the defender's commitment, and will rationally respond. In particular, the attacker first chooses a target to visit. If he observes a sensor at the target, the attacker then makes a second decision and determines to attack or not, based on the signal from the sensor. If the attacker chooses to not attack, both players get utility 0. The attacker will choose actions that maximize his utility.

6.2.2 Additional Challenges and Computational Hardness

We are interested in solving SEGs, by which we mean computing the *globally* optimal defender commitment consisting of the mixed strategy and signaling schemes. Without sensors in the game (i.e., $m = 0$), the problem can be easily solved by an $\mathcal{O}(n^2)$ algorithm called ORIGAMI (Kiekintveld, Jain, Tsai, Pita, Ordóñez, & Tambe, 2009). In this section, we illustrate the additional challenges due to the consideration of sensors by proving the NP-hardness of solving SEGs even in zero-sum cases. Then we formulate the problem using the multiple-LP approach (Conitzer & Sandholm, 2006).

Theorem 6.2.1. *Computing the optimal defender commitment is NP-hard even in zero-sum SEGs.*

Proof. We reduce from the *dominating set problem*. A dominating set for a graph G is a subset D of vertices such that every vertex is either in D or adjacent to a vertex in D. The dominating set

problem is to compute the size of a smallest dominating set for G. This problem is NP-hard even when G is a planar graph with maximum degree 3 (Garey & Johnson, 1979). We now reduce an arbitrary dominating set instance to our problem.

Given any graph G with n vertices, consider a zero-sum SEG instance with k patrollers and $m = n - k$ sensors. Let $\tau = 1$ and $U^d_+(i) = U^a_+(i) = 0, U^d_-(i) = -1 = -U^a_-(i)$ for every i. That is the defender receives utility 0 for successfully protecting a target and utility -1 for failing to protect a target. We now prove that G has a dominating set of size k *if and only if* the optimal defender utility is 0 in the constructed SEG. As a result, by solving SEGs, we can solve the dominating set problem by enumerating different k's, yielding the NP-hardness of solving SEGs.

$\Rightarrow$: If G has a dominating set D of size k, the defender can cover the k vertices in D with patrollers and cover all the remaining vertices with sensors. By definition, any vertex not in D, covered by a sensor, will be adjacent to a vertex in D and therefore is successfully protected. As a result, all vertices are successfully protected and the defender receives utility 0.

$\Leftarrow$: If the defender achieves utility 0, this must imply that each target is always successfully protected, i.e., either in state θ_+ or θ_{s+}. Otherwise, since attack failure has cost 0 to the attacker ($U^a_+(i) = 0$), the attacker will attack any target that is protected with probability $p < 1$, which would have resulted in a negative defender utility — a contraction. This implies that any pure strategy must aways protect every target, which means the vertices protected by the k patrollers must form a dominating set. □

A Formulation with Exponential-Size LPs

The main challenge of solving a SEG is its nature as a *bi-level* optimization problem since signaling schemes are built on top of the mixed strategy. We show that the problem can be formulated as multiple (exponential-size) LPs.

We first formulate the signaling process w.r.t. target i. For convenience, let $y_i = \mathbb{P}(e_i = \theta_{s+})$ and $z_i = \mathbb{P}(e_i = \theta_{s-})$ denote the marginal probabilities of states θ_{s+}, θ_{s-}, respectively. Thanks to Lemma 8, we can w.l.o.g. restrict to signaling schemes with two signals σ_1, σ_0 that result in the attacker best response of attacking and not attacking, respectively. Define variables $\pi^+_i = \pi_i(\theta_{s+}, \sigma_1) \in [0, y_i]$ and $\pi^-_i = \pi_i(\theta_{s-}, \sigma_1) \in [0, z_i]$. To guarantee that σ_1, σ_0 result in the desired attacker best responses, we need two constraints: $U^a_{\sigma_1}(\pi^+_i, \pi^-_i) = \pi^+_i \cdot U^a_+(i) + \pi^-_i \cdot U^a_-(i) \geq 0$ and $U^a_{\sigma_0}(\pi^+_i, \pi^-_i, y_i, z_i) = (y_i - \pi^+_i)U^a_+(i) + (z_i - \pi^-_i)U^a_-(i) \leq 0$. Under these constraints, the defender's expected utility from σ_1 is $U^d_{\sigma_1}(\pi^+_i, \pi^-_i) = \pi^+_i \cdot U^d_+(i) + \pi^-_i \cdot U^d_-(i)$. Recall that the defender utility from σ_0 is 0. Crucially, $U^a_{\sigma_1}, U^d_{\sigma_1}, U^a_{\sigma_0}$ are all *linear* functions of $\pi^+_i, \pi^-_i, y_i, z_i$.

With these representations of defender and attacker utilities from different signals, we are ready to present LPs to compute the optimal defender mixed strategy. For any fixed target t we

exhibit an LP that computes the optimal defender strategy, subject to visiting target t being the attacker's best response. Details are given in the following linear program with variables $\{p_{\mathbf{e}}\}_{\mathbf{e}\in\mathcal{E}}$ and $x_i, y_i, z_i, w_i, \pi_i^+, \pi_i^-$ for all $i \in [n]$.

$$
\begin{array}{lll}
\max & x_t U_+^d(t) + w_t U_-^d(t) + U_{\sigma_1}^d(\pi_t^+, \pi_t^-) & \\
\text{s.t.} & x_t U_+^a(t) + w_t U_-^a(t) + U_{\sigma_1}^a(\pi_t^+, \pi_t^-) \geq & \\
& \quad x_i U_+^a(i) + w_i U_-^a(i) + U_{\sigma_1}^a(\pi_i^+, \pi_i^-) & \forall\, i \neq t \\
& \sum_{\mathbf{e}\in\mathcal{E}: e_i = \theta_+} p_{\mathbf{e}} = x_i & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}: e_i = \theta_{s+}} p_{\mathbf{e}} = y_i & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}: e_i = \theta_{s-}} p_{\mathbf{e}} = z_i & \forall\, i \in [n] \\
& x_i + y_i + z_i + w_i = 1 & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}} p_{\mathbf{e}} = 1 & \\
& p_{\mathbf{e}} \geq 0 & \forall\, \mathbf{e} \in \mathcal{E} \\
& U_{\sigma_1}^a(\pi_i^+, \pi_i^-) \geq 0 & \forall\, i \in [n] \\
& U_{\sigma_0}^a(\pi_i^+, \pi_i^-, y_i, z_i) \leq 0 & \forall\, i \in [n] \\
& 0 \leq \pi_i^+ \leq y_i, \qquad 0 \leq \pi_i^- \leq z_i & \forall\, i \in [n]
\end{array}
\tag{6.6}
$$

In LP (6.6), variable $p_{\mathbf{e}}$ is the probability of pure strategy $\mathbf{e}$ and x_i, y_i, z_i, w_i are the marginal probabilities of different states. Program (6.6) is an LP since $U_{\sigma_1}^d, U_{\sigma_1}^a, U_{\sigma_0}^a$ are all linear functions. The last three sets of constraints guarantee that $\{\pi_i^+, \pi_i^-\}$ is a feasible signaling scheme at each target i. The first set of constraints enforce that visiting target t is an attacker best response. The remaining constraints define various marginal probabilities. It is easy to see that LP (6.6) computes the optimal defender commitment, subject to visiting target t being an attacker best response.

The optimal commitment can be computed by solving LP (6.6) for each t and picking the solution with maximum objective. A scalable algorithm for solving SEGs is given next.

6.2.3 A Branch-and-Price Approach

The challenge of solving SEGs are two-fold. First, LP (6.6) has exponentially many variables. Second, we have to solve LP (6.6) for each $t \in [n]$, which is very costly. In this section, we propose `SEGer` (`SEG`s `e`ngine with LP `r`elaxations) — a branch and price based algorithm — to solve SEGs. We omit the standard description of branch and price (see, e.g., (Barnhart, Johnson, Nemhauser, Savelsbergh, & Vance, 1998)) but highlight how `SEGer` instantiates the two key ingredients of this framework: (a) the column generation technique for solving LP (6.6) by developing scalable algorithms for the *slave problem*; (a) an efficient relaxation of LP (6.6) for branch-and-bound pruning. We will describe the column generation step first.

6.2.3.1 Column Generation & Scalable Algorithms for the Slave

Our goal is to efficiently solve the exponential-size LP (6.6). The idea of column generation is to start by solving a restricted version of LP (6.6), where only a small subset $\mathcal{E}' \subset \mathcal{E}$ of pure strategies are considered. We then search for a pure strategy $\mathbf{e} \in \mathcal{E} \setminus \mathcal{E}'$ such that adding $\mathbf{e}$ to $\mathcal{E}'$ improves the optimal objective value. This procedure iterates until no pure strategies in $\mathcal{E} \setminus \mathcal{E}'$ can improve the objective, which means an optimal solution is found. The restricted LP (6.6) is called the *master*, while the problem of searching for a pure strategy $\mathbf{e} \in \mathcal{E} \setminus \mathcal{E}'$ is referred to as the *slave* problem. The slave is derived from the dual program of LP (6.6), particularly, from the dual constraints corresponding to primal variable $p_{\mathbf{e}}$s. We omit its textbook derivation here (see, e.g., (Tambe, 2011) for details), and only directly describe the slave problem in our setting as follows.

Slave Problem: *Given different weights $\alpha_i, \beta_i, \gamma_i \in \mathbb{R}$ for each i, solve the following* weight maximization problem*:*

$$\textit{maximize}_{\mathbf{e}\in\mathcal{E}} \sum_{i:e_i=\theta_+} \alpha_i + \sum_{i:e_i=\theta_{s+}} \beta_i + \sum_{i:e_i=\theta_{s-}} \gamma_i. \tag{6.7}$$

We mention that $\alpha_i, \beta_i, \gamma_i$ in the slave are the optimal dual variables for the constraints that define x_i, y_i, z_i respectively in LP (6.6). The slave is an interesting resource allocation problem with multiple resource types (i.e., patrollers and sensors) which affect each other. Using a reduction from the dominating set problem, it is not difficult to prove the following.

Lemma 11. *The slave problem is NP-hard.*

Proof. The proof is similar to the proof of Theorem 6.2.1. By letting $\alpha_i = \beta_i = 1, \gamma_i = 0, \tau = 1$ and $m = n - k$, it is easy to show that the graph has an independent set of size k if and only if the slave problem has optimal objective value n. □

An MILP Formulation for the Slave

Next we propose a mixed integer linear program (MILP) formulation for the slave problem. Our idea is to use three binary vectors $\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3 \in \{0,1\}^n$ to encode for each target whether it is in state $\theta_+, \theta_{s+}, \theta_{s-}$ respectively. For example, target i is in state θ_{s+} if and only if $v_i^2 = 1$. The main challenge then is to properly set up linear (in)equalities over these vectors to precisely capture their constraints and relations.

The capacity for each resource type results in two natural constraints: $\sum_{i\in[n]} v_i^1 \leq k$ and $\sum_{i\in[n]}(v_i^2 + v_i^3) \leq m$. Moreover, since at most one resource is assigned to any target, we have

$v_i^1 + v_i^2 + v_i^3 \leq 1$ for each $i \in [n]$. Finally, we use the set of constraints $A^\tau \cdot \mathbf{v}^1 \geq \mathbf{v}^2$ to specify which vertices could possibly have state θ_{s+} (i.e., have a patroller within distance τ). To see that this is the correct constraint, we claim that no vertex in $\mathbf{v}^1$ is within distance τ to i *if and only if* $A_i^\tau \cdot \mathbf{v}^1 = 0$ where A_i^τ is the i'th row of A^τ. This is easy to verify for $\tau = 1$ and follows by induction for general τ. It turns out that these constraints are sufficient to encode the slave problem. Details are presented in MILP (6.8), whose correctness is summarized in Proposition 3. Here, $\alpha = (\alpha_1, \ldots, \alpha_n)^\top$ (β, γ defined similarly) and $\langle \mathbf{v}^1 \cdot \alpha \rangle$ is the inner product between $\mathbf{v}^1$ and α. The matrix $A \in \{0,1\}^{n \times n}$ is the adjacency matrix of G (but with ones on its diagonal), and A^τ is the τ'th power of A.

$$\begin{array}{lll} \text{maximize} & \langle \mathbf{v}^1 \cdot \alpha \rangle + \langle \mathbf{v}^2 \cdot \beta \rangle + \langle \mathbf{v}^3 \cdot \gamma \rangle & \\ \text{subject to} & \sum_{i \in [n]} v_i^1 \leq k & \\ & \sum_{i \in [n]} (v_i^2 + v_i^3) \leq m & \\ & v_i^1 + v_i^2 + v_i^3 \leq 1, & \text{for } i \in [n]. \\ & A^\tau \cdot \mathbf{v}^1 \geq \mathbf{v}^2 & \\ & \mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3 \in \{0,1\}^n & \end{array} \tag{6.8}$$

Proposition 3. *Let* $\{\widehat{\mathbf{e}}^1, \widehat{\mathbf{e}}^2, \widehat{\mathbf{e}}^3\}$ *be an optimal solution to MILP* (6.8)*. Then assigning* k *patrollers to vertices in* $\widehat{\mathbf{e}}^1$ *and* m *sensors to vertices in* $\widehat{\mathbf{e}}^2 + \widehat{\mathbf{e}}^3$ *correctly solves Slave* (6.7)*. Here, for a vector* $\mathbf{v} \in \{0,1\}^n$*, we say "*i *is in* $\mathbf{v}$*"* iff $v_i = 1$.

Proof. We prove that feasible solutions to MILP (6.8) precisely encode all pure strategies in $\mathcal{E}$, under the mapping that vertices in $\widehat{\mathbf{e}}^1$ have state s_+, vertices in $\widehat{\mathbf{e}}^2$ have state s_{s+} and vertices in $\widehat{\mathbf{e}}^3$ have state s_{s-}. As a result, the objective of MILP (6.8) equals the objective of the slave, yielding the desired conclusion.

First, any pure strategy in $\mathcal{E}$ must satisfy all constraints of MILP (6.8). To see this, we only need to argue the necessity of satisfying constraint $A^\tau \cdot \mathbf{v}^1 \geq \mathbf{v}^2$. Let A_i denote the i'th row of A. The non-zero entries in A_i specify all vertices within distance 1 from i. A standard inductive argument shows that the non-zero entries in the i'th row of A^τ, denoted by A_i^τ, are precisely all the vertices within distance τ to i. Let $\mathbf{v}^1$ denote the subset of vertices covered by patrollers. Then $A_i^\tau \cdot \mathbf{v}^1 > 0$ if and only if there is a vertex in $\mathbf{v}^1$ (i.e., covered by a patroller) that is within distance τ to i. Only such a vertex i can have $e_i^2 = 1$, and this is precisely captured by $A_i^\tau \cdot \mathbf{v}^1 \geq e_i^2$ for all i (i.e., $A^\tau \cdot \mathbf{v}^1 \geq \mathbf{v}^2$).

Conversely, a similar argument shows that any feasible solution to MILP (6.8) corresponds to a pure strategy in $\mathcal{E}$ by assigning k patrollers to vertices in $\widehat{\mathbf{e}}^1$ and m sensors to vertices in $\widehat{\mathbf{e}}^2 + \widehat{\mathbf{e}}^3$, concluding the proof of the proposition. □

A $\frac{1}{2}(1-\frac{1}{e})$-Approximation Algorithm for the Slave

Next, we design a polynomial-time algorithm to *approximately* solve the slave problem, which can be used to accelerate `SEGer`. Our algorithm is provably a $\frac{1}{2}(1-\frac{1}{e})$-approximation to the slave problem in zero-sum cases. The approximation guarantee relies on a special property of the slave for zero-sum SEGs, stated as follows, which unfortunately is not true in general. However, the algorithm can still be used as a good *heuristic* for solving general-sum SEGs. All the proofs in this part are deferred to Appendix B.

Lemma 12. *In zero-sum SEGs, the $\alpha_i, \beta_i, \gamma_i$ in Slave* (6.7) *are guaranteed to satisfy: $\alpha_i \geq \beta_i \geq \gamma_i \geq 0$ for any $i \in [n]$.*

Our algorithm originates from the following idea. The slave problem can be viewed as a two-step resource allocation problem. In the first step, a vertex subset T of size at most k is chosen for allocating patrollers; in the second step, a subset $I \subseteq [n] \setminus T$ of size at most m is chosen for allocating sensors. Our key observation is that given T, the second step of choosing I is easy. Let

$$T^N = \{i \mid i \notin T \text{ but } A^{\tau}_{i,j} > 0 \text{ for some } j \in T\}$$

denote the set of all vertices that are not in T but within distance τ to some vertices in T (interpreted as *neighbors* of T). With some abuse of notations, let $T^c = [n] \setminus (T \cup T^N)$ denote the set of remaining vertices. Notice that T, T^N, T^c are mutually disjoint. The following lemma illustrates how to pick the optimal set I, given T.

Lemma 13. *Given T, the second step of the slave (i.e., picking set I) simply picks the m vertices corresponding to the largest m weights in $\{\beta_i \mid i \in T^N\} \cup \{\gamma_i \mid i \in T^c\}$.*

Lemma 13 is true because when T is given, the weight of covering target i by a sensor is determined — either β_i if $i \in T^N$ or γ_i if $i \in T^c$. Thus the main difficulty of solving the slave problem lies at the first step, i.e., to find the allocation for patrollers. For convenience, let *operator* $\Sigma^m_{\max}(W)$ denote the sum of the largest m weights in weight set W. Utilizing Lemma 13, the objective value of the slave, parameterized by set T, can be viewed as a set function of T:

$$f(T) = \textstyle\sum_{i \in T} \alpha_i + \Sigma^m_{\max}\big(\{\beta_i \mid i \in T^N\} \cup \{\gamma_i \mid i \in T^c\}\big).$$

As a result, the slave problem can be re-formulated as a *set function maximization* problem:

$$\text{Slave Reformulation:} \qquad \max_{T \subset [n]: |T| \leq k} f(T).$$

The NP-hardness of the slave implies that there is unlikely to be a polynomial-time algorithm that maximizes $f(T)$ exactly. One natural question is whether $f(T)$ is *submodular*, since submodular maximization admits good approximation guarantees (Calinescu et al., 2011). Unfortunately, the answer turns out to be "No" (see Appendix B.2 for a counter example). Nevertheless, we show that maximizing $f(T)$ admits a constant approximation under certain conditions.

Theorem 6.2.2. *When $\alpha_i \geq \beta_i \geq \gamma_i \geq 0, \forall i \in [n]$, there is a poly-time $\frac{1}{2}(1-\frac{1}{e})$-approximate algorithm for the slave.*

A formal proof of Theorem 6.2.2 can be found in Appendix B; we only provide a proof sketch here. The key insight is that although $f(T)$ is not submodular, a *variant of* $f(T)$, defined below, can be proved to be submodular. Define

$$g(T) = \textstyle\sum_{i\in T} \alpha_i + \Sigma^m_{\max}\big(\{\beta_i \mid i \in T^N \cup T\} \cup \{\gamma_i \mid i \in T^c\}\big).$$

The only difference between $f(T)$ and $g(T)$ is that the weight set in the definition of $f(T)$ [resp., $g(T)$] contains β_is for any $i \in T^N$ [resp., $i \in T^N \cup T$]. Notice that $g(T)$ can be evaluated in polynomial time for any $T \subseteq [n]$.

Our algorithm, named `TailoredGreedy` (details in Algorithm 5), runs the greedy algorithm for maximizing $g(T)$ and then uses the output to construct a solution for the slave, i.e., for maximizing $f(T)$. The theorem can then be proved in two steps. First, we prove that $g(T)$ is monotone submodular. This requires a somewhat intricate proof with careful analysis of the function. Then we show that `TailoredGreedy` yields a $\frac{1}{2}(1-\frac{1}{e})$-approximation for the slave problem. The key step for proving this result is to establish the following relation between the functions $f(T)$ and $g(T)$: $f(T) \leq g(T) \leq 2f(T)$.

Algorithm 5: `TailoredGreedy`

Input: weights $\alpha_i, \beta_i, \gamma_i \in \mathbb{R}$ for any $i \in [n]$
Output: a pure strategy in $\mathcal{E}$

1: Initialization: $T = \emptyset$.
2: **for** t = 1 to k **do**
3: Compute $i^* = \arg\max_{i\in[n]\setminus T}[g(T \cup \{i\}) - g(T)]$.
4: Add i^* to T
5: **return** the pure strategy that covers the vertices in T with patrollers and covers the m vertices corresponding to the largest m weights in $\{\beta_i \mid i \in T^N\} \cup \{\gamma_i \mid i \in T^c\}$ with sensors.

6.2.3.2 LP Relaxation for Branch-and-Bound Pruning

Our goal of using branch-and-bound is to avoid solving LP (6.6) one by one for each t, which is too costly. The idea is to come up with an *efficiently computable* upper bound of LP (6.6) for

each t, so that once the best objective value among the solved LP (6.6)'s is larger than the upper bound of all the (yet) unsolved ones, we can safely claim that the current best solution is optimal without solving the remaining LPs. In this section, by properly relaxing LP (6.6) we obtain such an upper bound, which leads to significant speed-up in experiments.

The standard approach for finding relaxations in security games is to ignore scheduling constraints. Unfortunately, this does not work in our case since our security resources do not have scheduling constraints. The difficulty of our problem lies in characterizing marginal probabilities of different states in Θ. Our idea is to utilize the constraints in MILP (6.8). Observe that $\mathbf{v}^1, \mathbf{v}^2, \mathbf{v}^3$ in MILP (6.8) can be viewed as marginal vectors of a pure strategy for the states $\theta_+, \theta_{s+}, \theta_{s-}$ respectively. Recall that $\mathbf{x}, \mathbf{y}, \mathbf{z}$ in LP (6.6) are the marginal vectors of a mixed strategy $\mathbf{p}$ for state $\theta_+, \theta_{s+}, \theta_{s-}$ respectively. Therefore, the $\mathbf{x}, \mathbf{y}, \mathbf{z}$ of any pure strategy must satisfy the constraints in MILP (6.8) by setting $\mathbf{v}^1 = \mathbf{x}$, $\mathbf{v}^2 = \mathbf{y}$, $\mathbf{v}^3 = \mathbf{z}$. By linearity, the $\mathbf{x}, \mathbf{y}, \mathbf{z}$ of any mixed strategy must also satisfy these constraints. This results in a relaxation of LP (6.6) by substituting the constraints in LP (6.6) that define x_i, y_i, z_i with the constraints of MILP (6.8).

Proposition 4. *The following is a valid relaxation of LP* (6.6)*. Moreover, this relaxation results in a linear program with polynomially many variables and constraints.*

$$
\begin{array}{ll}
\sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_+} p_e = x_i, \forall i & \sum_{i\in[n]} x_i \le k \\
\sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_{s+}} p_e = y_i, \forall i & \sum_{i\in[n]} (y_i + z_i) \le m \\
\sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_{s-}} p_e = z_i, \forall i \quad \Longrightarrow & x_i + y_i + z_i \le 1,\ \forall i \\
\sum_{\mathbf{e}\in\mathcal{E}} p_{\mathbf{e}} = 1 & A^\tau \cdot \mathbf{x} \ge \mathbf{y} \\
p_{\mathbf{e}} \ge 0,\ \forall \mathbf{e} \in \mathcal{E} & \mathbf{x}, \mathbf{y}, \mathbf{z} \in [0,1]^n
\end{array}
$$

Relaxation: substitute left part in LP (6.6) with right part

6.2.4 Experiments

In this section, we experimentally test our model and algorithms. All LPs and MILPs are solved by CPLEX (version 12.7.1) on a machine with an Intel core i5-7200U CPU and 11.6 GB memory. All the game payoffs are generated via the covariant game model (Nudelman et al., 2004), which are widely adopted to test algorithms in security games. Let $\mu[a, b]$ denote the uniform distribution over the interval $[a, b]$. For any $i \in [n]$, we generate $U^d_+(i) \sim \mu[0, 10], U^d_-(i) \sim \mu[-10, 0], U^a_+(i) = cor \cdot U^d_+(i) + (1+cor)\cdot\mu[-10, 0]$ and $U^a_-(i) = cor \cdot U^d_-(i) + (1+cor)\cdot\mu[0, 10]$ where $cor \in [-1, 0]$ is a parameter controlling the *correlation* between the defender and attacker payoffs. The game is zero-sum when $cor = -1$. All general-sum games are generated with $cor = -0.6$ unless otherwise stated. The graph G is generated via the Erdös — Rényi random graph model.

Sensors Improve the Defender's Utility

Figure 6.3 shows the comparison of the defender utility under different scenarios. All data points in Figure 6.3 are averaged over 30 random instances and each instance has 30 targets.

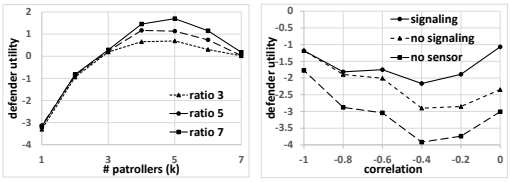


Figure 6.3: Utility comparison

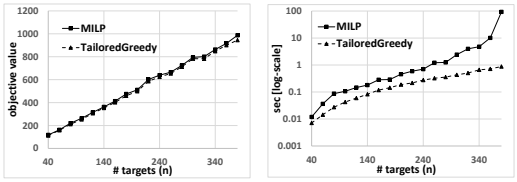
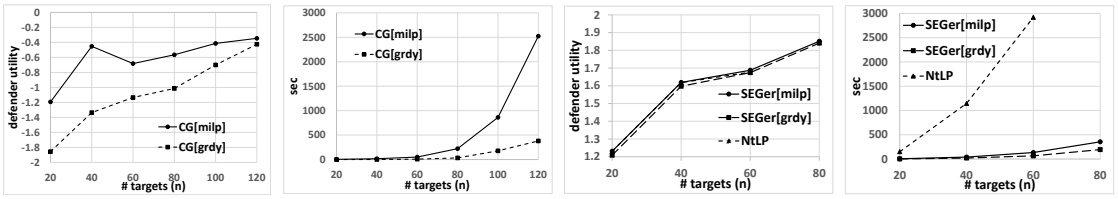


Figure 6.4: `TailoredGreedy` vs. MILP



(a) Zero-sum, def utility (b) Zero-sum, run time (c) General-sum, def utility (d) General-sum, run time

Figure 6.5: Utility comparison and scalability test of different algorithms for solving general-sum and zero-sum SEGs.

The left panel of Figure 6.3 compares the following scenarios. The defender has a fixed budget that equals the total cost of 7 patrollers, and the cost of a patroller may equal the cost of 3 or 5 or 7 sensors (corresponding to ratio 3, ratio 5 and ratio 7 line, respectively). The x-axis coordinate k means that the defender gets k patrollers and $ratio \times (7 - k)$ sensors; the y-axis is the defender utility. The figure demonstrates that a *proper combination* of patrollers and sensors results in better defender utility than just having patrollers (i.e., $k = 7$). This is the case even when the cost ratio is 3. The figure also shows that many sensors with few patrollers will not perform well, either. Therefore, the number of patrollers and sensors need to be properly balanced in practice.

The right panel of Figure 6.3 compares the defender utility in three different models: 1. `signaling` — SEG model; 2. `no signaling` — SEG model but assuming sensors do not strategically signal; 3. `no sensor` — classical security games. Both `signaling` and `no signaling` have 4 patrollers and 10 sensors while `no sensor` has 6 patrollers with no sensors (i.e., cost ratio between the patroller and sensor is 5). The x-axis is the correlation parameter of the general-sum games. The graph G used in this figure is a cycle graph motivated by the protection of the border of conservation parks as in our previous illustrative example. The figure shows that `signaling` results in higher utility than `no signaling`, demonstrating the benefit of using strategic signaling in this setting. Such a benefit decreases as the game becomes closer to being

zero-sum (i.e., *cor* tends to -1). This is as expected since signaling does not help in zero-sum settings due to its strict competition — any information to the attacker will benefit the attacker, and thus hurt the defender in a zero-sum setting. Both `signaling` and `no signaling` result in a stably higher utility that `no sensor` regardless of players' payoff correlation.

`TailoredGreedy` vs. MILP

In Figure 6.4, we compare the performances of MILP (6.8) and `TailoredGreedy` on solving just the *slave* problem. Notice that that running time in the right panel is in *logarithmic scale*. Each data point is an average over 15 instances with randomly generated $\alpha_i \geq \beta_i \geq \gamma_i \geq 0$ for each $i \in [n]$. Figure 6.4 shows that `TailoredGreedy` achieves only slightly worse objective value than MILP but is much more scalable. The scalability superiority of `TailoredGreedy` becomes prominent for larger instances ($n \geq$280) where MILP starts to run in exponential time while `TailoredGreedy` is a polynomial time algorithm.

Game Solving: Utility & Scalability Comparisons

Finally, we compare the performance of different algorithms in solving SEGs in Figure 6.5. Since zero-sum SEGs can be formulated as a single LP, which can then be solved by column generation. We compare two algorithms in this case: `CG[milp]` — column generation with MILP (6.8) for the slave; `CG[grdy]` — column generation with `TailoredGreedy` for the slave. Note that `CG[milp]` is optimal while `CG[grdy]` is not optimal since it uses an approximate algorithm for the slave.[3] Figure 6.10(a) shows that our algorithms can solve zero-sum SEGs with $80 \sim 100$ targets (depending on the algorithm) within 10 minutes. `CG[grdy]` achieves less utility than `CG[milp]`, but is more scalable (exact calculations show that `CG[grdy]` is at least 6 times faster). The utility gap between `CG[milp]` and `CG[grdy]` becomes smaller as n grows, while their running time gap becomes larger. This suggests that it might be more desirable to use `CG[milp]` for small instances and `CG[grdy]` for large instances if some utility loss is acceptable.

For general-sum SEGs (Figures 6.10(d) and 6.10(c)), we consider three algorithms: 1. `SEGer[milp]` — `SEGer` using MILP for column generation; 2. `SEGer[grdy]` — `SEGer` using `TailoredGreedy` for column generation; 3. `NtLP` — solving LP (6.6) one by one for each t without branch and bound. Though `SEGer[grdy]` is not optimal, it achieves close-to-optimal objective value in this case and runs faster than `SEGer[milp]` (roughly half of the running time of `SEGer[milp]`). On the other hand, both `SEGer[milp]` and

[3] We also implemented the algorithm that uses `TailoredGreedy` first and then switches to MILP when `TailoredGreedy` does not improve the objective. However, this approach seems to not help in our case and results in the same running time as `CG[milp]`. Thus we do not present it here.

`SEGer[grdy]` are much more scalable than `NtLP`. In fact, the running time for solving a general-sum SEG by `SEGer[milp]` is only slightly more than the running time of solving a zero-sum SEG of the same size, which demonstrates the significant advantage of our branch and price algorithm.

6.3 Exploiting Informational Advantage in Bayesian Stackelberg Games

The previous two sections developed Stackelberg security game models which allow the defender to commit not only to a distribution over actions, but also to a scheme for stochastically signaling information about these actions to the attacker. This can result in higher utility for the defender. In this section, we extend this methodology to general Bayesian games, in which either the leader or the follower or both have payoff-relevant private information. 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 (Conitzer & Sandholm, 2006; Letchford, Conitzer, & Munagala, 2009), 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.

6.3.1 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.

	$F_\emptyset$	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 6.6: Payoff matrices for followers of different types

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 6.6, 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 (Conitzer & Sandholm, 2006). Nevertheless, by exploiting the special structure in this example, it is easy to show that any mixed strategy that puts at least $2/3$ 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.

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

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,[4] 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 before 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

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

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, and 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.

6.3.2 Single Leader Type, Multiple Follower Types

We now generalize the example in Section 6.3.1 and consider how the leader's additional ability of committing to a signaling scheme changes the game and the computation. We start with the 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 in normal form 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_{ij}^θ $[b_{ij}^\theta]$ 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.[5] 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 normal-form BSG (Conitzer & Sandholm, 2006). A later result strengthened the hardness to approximation — no polynomial time algorithm can give a non-trivial approximation ratio in general unless P=NP (Letchford et al., 2009).

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

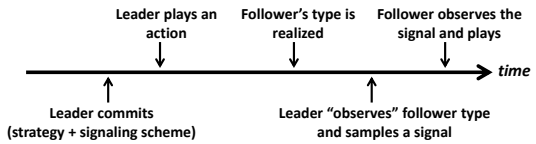


Figure 6.7: Timeline of the BSG with multiple follower types.

We consider a richer model where the defender can commit not only to a mixed strategy but also to a signaling scheme of partially revealing information regarding the action she is currently playing, i.e., the realized sample of the leader's 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 *randomized* 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

$$(\mathbf{x}, \varphi) \text{ where } \mathbf{x} \in \Delta_m;\ \varphi : \Theta \times [m] \xrightarrow{rnd} \Sigma \tag{6.9}$$

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 6.7 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 in 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 6.3.1, 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 genuinely learning their types, e.g., the follower's type may be defined by their location, in which case the leader can send signals using location-specific devices such as physical signs or radio transmission — this fits our model just as well. We will elaborate on one such example when discussing security games.

6.3.2.1 Normal-Form Games

We first consider the case where the leader can explicitly observe the follower's type, and thus can signal differently to different follower types. Like in the Bayesian persuasion model, we can

w.l.o.g. focus on direct signaling schemes that use at most n signals with signal σ_j recommending action $j \in [n]$ to the follower. As a result, we assume that $\Sigma = \{\sigma_j\}_{j\in[n]}$.

Theorem 6.3.1. *The optimal leader policy can be computed in $poly(m, n, |\Theta|)$ time by linear programming.*

Proof. Let $\mathbf{x} = (x_1, \ldots, x_m) \in \Delta_m$ be the leader's mixed strategy to commit to. A direct 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, $p_{ij}^{\theta} = x_i \cdot \varphi(j|i, \theta)$ 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 $\{p_{ij}^{\theta}\}_{i\in[m],j\in[n],\theta\in\Theta}$.

$$
\begin{array}{lll}
\text{maximize} & \sum_{\theta\in\Theta} \lambda_\theta \sum_{ij} p_{ij}^{\theta} a_{ij}^{\theta} & \\
\text{subject to} & \sum_{j=1}^{n} p_{ij}^{\theta} = x_i, & \text{for } i \in [m], \theta \in \Theta. \\
& \sum_{i=1}^{m} p_{ij}^{\theta} b_{ij}^{\theta} \geq \sum_{i=1}^{m} p_{ij}^{\theta} b_{ij'}^{\theta}, & \text{for } \theta, j \neq j'. \\
& \sum_{i=1}^{m} x_i = 1 & \\
& p_{ij}^{\theta} \geq 0, & \text{for all } i, j, \theta.
\end{array}
\tag{6.10}
$$

.

The first set of constraints mean that the summation of probability mass p_{ij}^{θ} — 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. □

Given any game G, let $U_{sig}(G)$ be the leader's expected utility by taking the optimal leader policy computed by LP (6.10). 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 5. *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 5 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 Appendix C.1

Remark: As we mentioned earlier, computing the optimal mixed strategy (assuming no signaling) to commit to is NP-hard to approximate within any non-trivial ratio (Conitzer & Sandholm, 2006; Letchford et al., 2009). 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 (Daskalakis et al., 2006; Chen et al., 2009)) and correlated equilibria (easy in fairly general settings (Papadimitriou & Roughgarden, 2008; Jiang & Leyton-Brown, 2011)).

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 interest 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 6.3.1). 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.

Note that focusing on direct signaling schemes is still without loss of generality 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, the leader 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 6.3.2. *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. We still use variables $\mathbf{x} \in \Delta_m$ and $\{p_{ij}^{\theta}\}_{i\in[m],j\in[n],\theta\in\Theta}$ to capture the leader's policy. Then $\alpha_j^{\theta} = \sum_{i=1}^{m} p_{ij}^{\theta}$ 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

$\frac{1}{\alpha_j^\beta}\sum_{i=1}^m p_{ij}^\beta b_{ij'}^\theta$. Therefore, the follower's action will be $\arg\max_{j'}\frac{1}{\alpha_j^\beta}\sum_{i=1}^m p_{ij}^\beta b_{ij'}^\theta$ with expected utility $\max_{j'}\frac{1}{\alpha_j^\beta}\sum_{i=1}^m p_{ij}^\beta b_{ij'}^\theta$. As a result, the expected utility for the follower of type θ, but misreporting type β, is

$$U(\beta;\theta)=\sum_{j=1}^n\left[\alpha_j^\beta\times\max_{j'}\frac{1}{\alpha_j^\beta}\sum_{i=1}^m p_{ij}^\beta b_{ij'}^\theta\right]=\sum_{j=1}^n\left[\max_{j'}\sum_{i=1}^m p_{ij}^\beta b_{ij'}^\theta\right].$$

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 $\max_{j'}\sum_{i=1}^m p_{ij}^\theta b_{ij'}^\theta=\sum_{i=1}^m p_{ij}^\theta b_{ij}^\theta$, i.e., the recommended action j by σ_j is indeed the follower's best response when the follower has type θ, we have

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

Therefore, incorporating the above constraints to LP (6.10) gives the following optimization program which computes an optimal incentive compatible leader policy.

$$\begin{array}{lll}
\text{maximize} & \sum_{\theta\in\Theta}\lambda_\theta\sum_{ij}p_{ij}^\theta a_{ij}^\theta & \\
\text{subject to} & \sum_{j=1}^n p_{ij}^\theta=x_i, & \text{for all } i,\theta.\\
& \sum_{i=1}^m p_{ij}^\theta b_{ij}^\theta\geq\sum_{i=1}^m p_{ij}^\theta b_{ij'}^\theta, & \text{for } j\neq j'.\\
& \sum_{j=1}^n\sum_{i=1}^m p_{ij}^\theta b_{ij}^\theta\geq & \\
& \quad\sum_{j=1}^n\left[\max_{j'}\sum_{i=1}^m p_{ij}^\beta b_{ij'}^\theta\right], & \text{for } \beta\neq\theta.\\
& \sum_{i=1}^m x_i=1 & \\
& p_{ij}^\theta\geq 0, & \text{for all } i,j,\theta.
\end{array}\tag{6.11}$$

Notice that $\sum_{j=1}^n\left[\max_{j'}\sum_{i=1}^m p_{ij}^\beta b_{ij'}^\theta\right]$ 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., (Boyd & Vandenberghe, 2004)). □

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 (6.11). The following theorem captures the utility ranking of the different models.

Proposition 6 (Utility Ranking)**.**

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

Proof. The first inequality holds because any feasible solution to Program (6.11) must be feasible to LP (6.10). 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., 2017). 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 (Kolotilin et al., 2017), 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 (6.11) is given *a-priori* instead of being designed.

6.3.2.2 Security Games

We now consider Bayesian *Security Games*, a particular type of Stackelberg game 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. 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 (Pita, Jain, Marecki, Ordóñez, Portway, Tambe, Western, Paruchuri, & Kraus, 2008b). In wildlife patrolling, the type of an attacker could be the species the attacker is interested in (Fang, Stone, & Tambe, 2015). 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 $U_{c/u}^{d/a}(i|\theta)$ 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.

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 that this is not possible here. It turns out that the complexity of the problem depends on how many targets an attacker is interested in. We say that an attacker of type θ is *not interested* in attacking target i if there exists j

such that $U_u^a(i|\theta) < U_c^a(j|\theta)$. 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. Unfortunately, this is *not* the case.

Proposition 7. *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 7 requires a slight modification of a similar proof in (Li, Conitzer, & Korzhyk, 2016), and is provided in Appendix C.2 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 (Yin et al., 2012) 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.[6]

Proposition 8. *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 8 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 Appendix C.2. 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 in (Xu, Jiang, Sinha, Rabinovich, Dughmi, & Tambe, 2015), and our proof does not naturally extend to this setting.

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

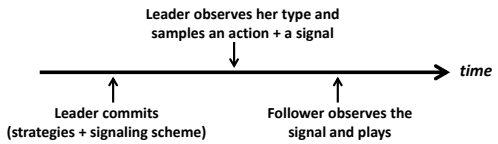


Figure 6.8: Timeline of the BSG with multiple leader types

Our next result shows that the restriction in Proposition 8 is almost necessary for efficient computation, as evidence of computational hardness manifests itself when we slightly go beyond the condition there.

Proposition 9. *The defender oracle problem* [7] *is NP-hard (both with and without type-reporting IC constraints), even when each type of attacker is interested in two targets.*

6.3.3 Multiple Leader Types, Single Follower Type

Similarly to Section 6.3.2, we still start with the normal-form Bayesian Stackelberg Game, but with multiple *leader* types and a single *follower* type. Following the notation in Section 6.3.2, 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 (Conitzer & Sandholm, 2006). 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

$$(\{\mathbf{x}^\theta\}_{\theta\in\Theta}, \varphi) \text{ where } \mathbf{x}^\theta \in \Delta_m;\ \varphi : \Theta \times [m] \xrightarrow{rnd} \Sigma \tag{6.12}$$

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 6.8 illustrates the timeline of the game.

[7] *The optimal policy can be computed by an LP with exponential size. The defender oracle is a main subroutine for solving the dual of the LP. See Appendix C.2 for a derivation of the defender oracle and proof of the hardness.*

6.3.3.1 Normal-Form Games

We focus on direct signaling schemes and assume $\Sigma = \{\sigma_1, \ldots, \sigma_n\}$ where σ_j is a signal recommending action j to the follower.

Theorem 6.3.3. *The optimal* leader policy *defined in Formula* (6.12) *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 $\{p^\theta_{ij}\}_{i\in[m],j\in[n],\theta\in\Theta}$.[8]

$$\begin{array}{lll} \text{maximize} & \sum_{\theta\in\Theta} \lambda_\theta \sum_{ij} p^\theta_{ij} a^\theta_{ij} & \\ \text{subject to} & \sum_{i=1}^m \sum_{j=1}^n p^\theta_{ij} = 1, & \text{for } \theta \in \Theta. \\ & \sum_{i,\theta} \lambda_\theta p^\theta_{ij} b^\theta_{ij} \geq \sum_{i,\theta} \lambda_\theta p^\theta_{ij} b^\theta_{ij'}, & \text{for } j \neq j'. \\ & p^\theta_{ij} \geq 0, & \text{for all } i, j, \theta. \end{array} \tag{6.13}$$

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 $(\{\mathbf{x}^\theta\}_{\theta\in\Theta}, \varphi)$. □

6.3.3.2 Security Games

We now again consider the security game setting. We have shown in Section 6.3.2 that, when there are multiple follower types, the polynomial-time solvability of BSGs does not extend to even the simplest security game setting. 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 show below.

We still adopt the notations from Section 6.3.2.2, except that θ is now the defender's private type. We further allow scheduling constraints in the defender's resource allocation. Recall that the set of defender pure strategies $\mathcal{E}$ and the set of marginal probabilities $\mathcal{P}$ have been described in Section 2.2.1. It was shown previously that if the defender's best response problem can be solved in polynomial time, then the Strong Stackelberg equilibrium can also be computed in polynomial time (Jain et al., 2010; Xu, 2016). We now establish an analogous result for BSG with signaling.

Theorem 6.3.4. *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.*

[8] When $|\Theta| = 1$, the game degenerates to a Stackelberg game without uncertainty of player types, and LP (6.13) degenerates to a linear program that computes the Strong Stackelberg equilibrium (Conitzer & Korzhyk, 2011).

Proof. First, observe that LP (6.13) does not obviously extend to security game settings because the number of leader pure strategies 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 (6.13)), and finally picks a mixed strategy that depends on both θ and σ_j (i.e., the vector $(p_{1j}^{\theta}, p_{2j}^{\theta}, \ldots, p_{mj}^{\theta})$ normalized by the factor $\sum_{i=1}^{m} p_{ij}^{\theta}$ in LP (6.13)).

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

$$\sum_{\theta} \left(\frac{\lambda_{\theta} p_j^{\theta}}{\alpha_j} \cdot \left[\mathbf{x}_j^{\theta}(j') U_c^a(j'|\theta) + \left(1 - \mathbf{x}_j^{\theta}(j')\right) U_u^a(j'|\theta) \right] \right) \tag{6.14}$$

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,

$$\begin{aligned} & AttU(j, j') \\ = \;& \alpha_j \times \text{Equation (6.14)} \\ = \;& \textstyle\sum_{\theta} \left(\lambda_{\theta} p_j^{\theta} \mathbf{x}_j^{\theta}(j') U_c^a(j'|\theta) + \left[\lambda_{\theta} p_j^{\theta} - \lambda_{\theta} p_j^{\theta} \mathbf{x}_j^{\theta}(j') \right] U_u^a(j'|\theta) \right) \end{aligned}$$

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 $\{p_j^{\theta}\}_{j \in [n], \theta \in \Theta}$.

$$\begin{array}{lll} \text{maximize} & \sum_j DefU(j, j) & \\ \text{subject to} & AttU(j, j) \geq AttU(j, j'), & \text{for } j \neq j'. \\ & \sum_j p_j^{\theta} = 1, & \text{for } \theta \in \Theta. \\ & \mathbf{x}_j^{\theta} \in \mathcal{P}, & \text{for } j, \theta. \\ & p_j^{\theta} \geq 0, & \text{for } j, \theta. \end{array} \tag{6.15}$$

The optimization program (6.15) 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 (6.15) is $\mathbf{x}_j^\theta \in \mathcal{P}$, which now becomes $\mathbf{y}_j^\theta \in p_j^\theta \mathcal{P}$ where both p_j^θ and $\mathbf{y}_j^\theta$ are variables. Here $p\mathcal{P}$ denotes the polytope $\{px : x \in \mathcal{P}\}$ for any given p. This turns out to still be a convex constraint, and behaves nicely as long as the polytope $\mathcal{P}$ behaves nicely.

Lemma 14 (Polytope Transformation)**.** *Let $\mathcal{P} \subseteq \mathbb{R}^n$ be any* bounded *convex set. Then the following hold:*

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

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

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

The proof of Lemma 15 is standard, and is deferred to Appendix C.3. We note that the restriction that $\mathcal{P}$ is bounded is important; otherwise, some conclusions do not hold, e.g., Property 2. Fortunately, the polytope $\mathcal{P}$ of mixed strategies is bounded. Therefore, using Lemma 15, we can rewrite Quadratic Program (6.15) as the following linear program.

$$\begin{array}{lll} \text{maximize} & \sum_j DefU(j,j) & \\ \text{subject to} & AttU(j,j) \geq AttU(j,j'), & \text{for } j \neq j'. \\ & \sum_j p_j^\theta = 1, & \text{for } \theta \in \Theta. \\ & (\mathbf{y}_j^\theta, p_j^\theta) \in \widetilde{\mathcal{P}}, & \text{for } j, \theta. \\ & p_j^\theta \geq 0, & \text{for } j, \theta. \end{array} \tag{6.16}$$

Program (6.16) is linear because $AttU(j,j')$ and $DefU(j,j)$ are linear in p_j^θ and $\mathbf{y}_j^\theta$, and moreover, $(\mathbf{y}_j^\theta, p_j^\theta) \in \widetilde{\mathcal{P}}$ are essentially linear constraints due to Lemma 15 and the fact that $\mathcal{P}$ is a polytope in security games. Furthermore, LP (6.16) has a compact representation as long as the polytope of realizable mixed strategies $\mathcal{P}$ has one. In this case, LP (6.16) can be solved explicitly. More generally, by standard techniques from convex programing, we can show that the separation oracle for $\mathcal{P}$ 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 $\mathcal{P}$ can be found in $poly(n)$ time. By Lemma 15, $\widetilde{\mathcal{P}}$ then admits a $poly(n)$ time separation oracle, so LP (6.16) can solved in $poly(n, |\Theta|)$ time. The proof is not particularly insightful and a similar argument can be found in (Xu, Fang, Jiang, Conitzer, Dughmi, & Tambe, 2014). So we omit the details here. □

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 and only privately observes her type, our model degenerates to the *Bayesian Persuasion* (BP)

model of (Kamenica & Gentzkow, 2011). Our model generalizes the BP model to the case where the sender has both actions and 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. (Rabinovich et al., 2015). 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. (Rabinovich et al., 2015) were able to efficiently solve for the case with unconstrained identical security resources. Our Theorem 6.3.4 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.

6.3.4 Experiments

We mainly present the comparison of the models discussed in Section 6.3.2 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 (Nudelman et al., 2004). For any i, j, and Θ, we independently set a_{ij}^{θ} equal to a random integer in the range $[-5, 5]$ for each i, j, θ. The probabilities $\{\lambda_\theta\}_{\theta \in \Theta}$ are 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 $\alpha = 0$ the game is zero-sum, while $\alpha = 1$ means completely uncorrelated 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.

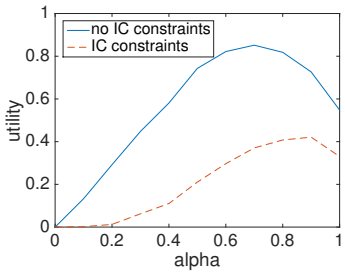
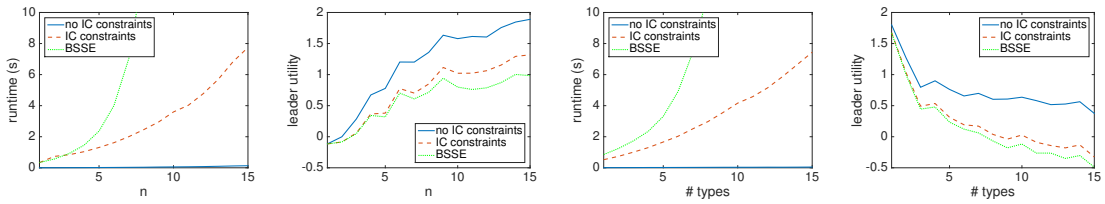


Figure 6.9: Extra utility gained by the leader from signaling.

We first consider the value of signaling for different values of α chosen from the set $\{0, 0.1, 0.2, \ldots, 1\}$. For these simulations, we fix $m = n = 10$ and $|\Theta| = 5$. Figure 6.9 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, by Proposition 5. 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, and 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.



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

Figure 6.10: Runtime and utility comparisons by varying the number of actions n and the number of types $|\Theta|$ for the three different models in the case of multiple follower types.

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 6.10(a) and 6.10(b), we hold the number of follower types constant ($|\Theta| = 5$) and vary $m = n$ between 1 and 15. In Figures 6.10(c) and 6.10(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 6.10(a) and 6.10(c).

In both figures of leader utility, the differences of the leader's utility among the models are as indicated by Proposition 6. 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

[9] To compute the BSSE, we implement the state-of-art algorithm DOBBS, a mixed integer linear program as formulated in (Paruchuri, Pearce, Marecki, Tambe, Ordonez, & Kraus, 2008).

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 (Conitzer & Sandholm, 2006). 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.

Part III

Dealing with Information Leakage

Chapter 7

Real-World Motivation and Two Illustrative Examples

In this chapter, we describe two concrete examples motivated from real-world domains that illustrate the issue of information leakage in security games. Our examples show that such leakage may cause significant loss to the defender if not addressed properly.

7.1 Motivating Example I: Information Leakage in Air Marshal Scheduling

Our first example considers the problem of scheduling federal air marshals to protect flights (Tsai, Rathi, Kiekintveld, Ordonez, & Tambe, 2009). With more than 30,000 commercial flights per day in the United States airspace but only a limited number of air marshals, the Federal Air Marshal Service (FAMS) can only cover a small portion of flights and has to schedule air marshals in a randomized fashion. Naturally, such randomization needs to be intelligently designed based on the risk and importance of different flights. To assist in this large-scale scheduling process, a software assistance called Intelligent Randomization in Scheduling (IRIS) has been deployed and is currently in use by the FAMS (Tsai et al., 2009).

Figure 7.1: A tweet that leaks information

Figure 7.2: A round-trip schedule with information leakage.

When designing IRIS, a crucial assumption made was that the attacker could only observe the defender's mixed strategy but was not able to observe, even partially, the defender's pure strategy. However, this assumption may fail in practice. The realized protection status of some flights may leak out to the adversary due to various reasons, e.g., even an unintentional tweet (Figure 7.1). Since the air marshal's schedule is usually a round trip or even a multi-way trip, if the adversary knows the protection status of a certain flight, he can infer the protection status of return flights (see Figure 7.2). It turns out that such information leakage may cause a significant loss to the defender if not addressed carefully.

To illustrate this vulnerability, we consider a simple example in which the FAMS needs to protect four flights — two from LAX to ORD (denoted as A_1, A_2) and two return flights from ORD to LAX (denoted as B_1, B_2). There is only one air marshal available. The flights are depicted in the left panel of Figure 7.3. We assume that any outbound flight can form a round trip with any return flight — i.e., their arrival and departure times are compatible. Assume that, due to the different importances of the flights, the desired marginal protection probability is $2/3$ for flights A_1 and B_1 and is $1/3$ for flights A_2 and B_2 when there is no information leakage.

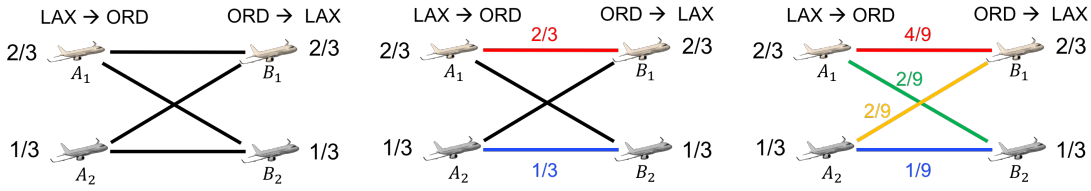


Figure 7.3: Desired marginal protection probabilities and two different mixed strategies to implement the marginals.

There have been different algorithms (Tsai et al., 2009; Kiekintveld et al., 2009; Jain et al., 2010) developed to efficiently compute the optimal defender mixed strategy — i.e., a distribution over the air marshal's schedules — for the air marshal scheduling problem. However, they all assume that the attacker does not observe the defender's realized pure strategy. Under this assumption, it does not matter how we implement the marginal protection probabilities. One computational challenge here is the exponential explosion of the total number of pure strategies due to the combinatorial structure of the defender's strategy. To overcome this challenge, most efficient algorithms are designed to implement the desired marginal protection probabilities by randomizing over as few schedules as possible. This is also the reason that they are efficient — if the algorithm randomizes over too many schedules, it is not efficient any more.

In this example, most efficient algorithms will output the strategy depicted in the middle panel of Figure 7.3. That is, the air marshal will take the red round trip with probability $2/3$ and the

blue round trip with probability $1/3$. It is easy to verify that this mixed strategy implements the desired marginal protection probabilities.

Unfortunately, such a mixed strategy with small support may be extremely vulnerable to information leakage. In fact, in this example, the attacker can completely uncover the air marshal's schedule by observing the protection status of just one flight — actually, any flight. This is because this mixed strategy creates too much correlation among flights. For example, consider that the adversary can observe the protection status of flight A_2 (e.g., because he sees a tweet about an air marshal as in Figure 7.1). Now, if flight A_2 is protected, this implies that the air marshal is taking the blue trip; therefore flight B_1 will not be protected later. Therefore, the attacker will have some time to plan an attack on B_1. On the other hand, if flight A_2 is not protected, this implies that the air marshal is taking the red trip and flight B_2 will not protected later. In either case, the attacker can always identify a completely uncovered flight to attack.

This example illustrates that previous algorithms can be extremely vulnerable to information leakage. In particular, these algorithms save running time by generating small-support mixed strategies which tends to introduce strong correlation among flights and make the strategies vulnerable to leakage. This seems to create a dilemma between time efficiency and robustness to leakage. The next two chapters will illustrate how we can overcome such a dilemma.

In this particular example, one possible way of overcoming the vulnerability to information leakage is to design a different schedule distribution that achieves the same marginal protection probabilities but has much less correlation among flights. The right panel of Figure 7.3 depicts one such implementation. It is easy to verify that the distribution implements the given marginal probabilities. However, it has much less correlation among flights. For example, even if the attacker knows that flight A_2 is protected, he is still uncertain whether B_1 or B_2 will be protected later since both the orange and blue trip are possible. We note that the distribution of the air marshal's schedule here is the max-entropy distribution subject to achieving the given marginal distributions. As we will see, the max-entropy distribution turns out to be a natural and useful choice in the presence of information leakage.

Remark. One might wonder how much information the attacker needs in order to infer the protection status of a flight from the status of another. In particular, would this require the attacker to know the whole pure strategy — i.e., the probability of each pure strategy, which may be unrealistic for the attacker to know? For such inference, it is enough for the attacker to just know the correlation of the protection status for each pair of flights. In fact, if the attacker has a more specific idea about which particular target to observe and which to attack, he would only need to know the correlation among these two flights.

7.2 Motivating Example II: Information Leakage in Patrol Route Design

Our second example considers the design of randomized patrol routes for rangers in order to combat poachers' poaching activity. Information leakage is also a very important concern in this setting due to the poacher's partial observation of rangers' patrol routes.

There have been many works optimizing the design of randomized patrol routes under different game settings and player rationality models (Fang et al., 2015; Nguyen, Delle Fave, Kar, Lakshminarayanan, Yadav, Tambe, Agmon, Plumptre, Driciru, Wanyama, et al., 2015; Fang et al., 2016a). The essential charge of all these works — regardless of which model or algorithm is adopted — is to create *unpredictability via randomization*. However, despite te fact that there are usually a huge number of patrol routes to choose from, most efficient algorithms tend to randomize over as few routes as possible, as we illustrated in Section 7.1. Such small-support mixed strategies usually result in high correlation among targets. This opens the door for the poacher to use his partial observation to infer the rangers' upcoming patrol directions. In fact, such an issue of information leakage has been a widely known concern in wildlife conservation (Nyirenda & Chomba, 2012; Moreto, 2013).

To be more concrete, we now give an example illustrating how the attacker can first observe the protection status of a single target and then utilize correlations to infer the protection of other targets. To do so, the attacker does *not* even need full knowledge of the defender's mixed strategy. Instead he only needs to know the correlations among the observed targets and his targets.

Consider the problem of designing rangers' patrolling routes within a fixed time period, say, a day. This is usually modeled by discretizing the area into cells as well as discretizing the time. At the top of Figure 7.4, we depict a concrete example with 4 cells to be protected at 3 time layers — morning, noon and afternoon. The numbers around each cell are the desired marginal coverage probabilities for each cell at each time. For simplicity, we assume that as time goes by, the ranger can move from any cell to any other one without constraints. The defender has 2 rangers, and seeks to randomize their patrolling to achieve the required marginal probabilities.

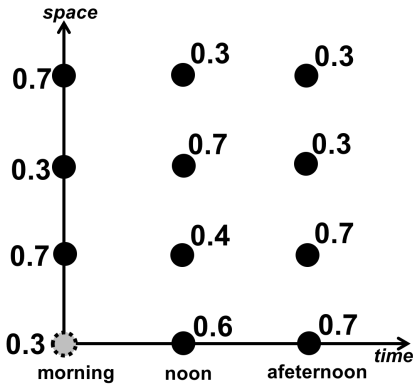


Figure 7.4: An example with four cells to be protected within three time layers.

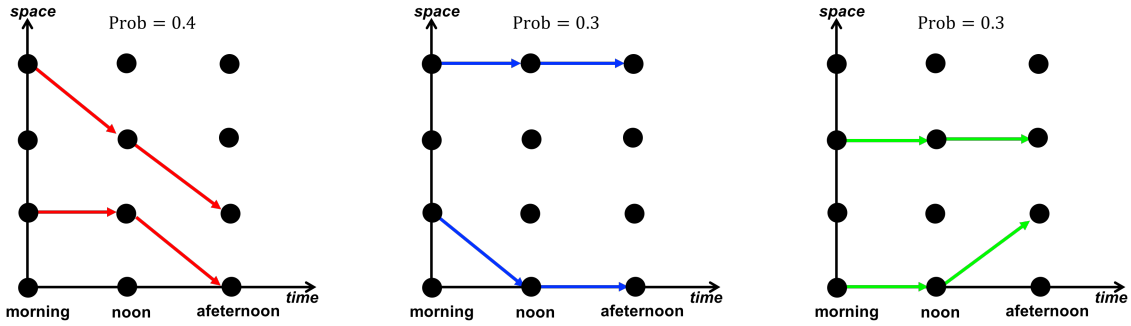


Figure 7.5: One mixed strategy that implements the marginals in Figure 7.4

Naturally, there are many different ways to implement this mixed strategy. As we mentioned before, classic algorithms strive to compute a mixed strategies of small support. In this example, previous algorithms tend to output the mixed strategy, as depicted in Figure 7.5, that randomizes over the three pure strategies.

Unfortunately, such an implementation is extremely vulnerable to the attacker's partial surveillance. For example, if the attacker can surveil the protection status of the bottom cell in the morning (i.e., the one with gray color and dashed boundary in Figure 7.4) and prepare an attack in the afternoon, he can always find a completely uncovered cell to attack. Specifically, if the dashed cell is covered, this means the third strategy in Figure 7.5 is deployed and the attacker can find two completely uncovered cells in the afternoon; Otherwise, either the first or the second strategy is deployed; thus the second-from-the-top cell will be uncovered in the afternoon for sure. To sum up, the attacker can successfully identify uncovered cells in the afternoon by monitoring only *one* cell in the morning.

7.3 The Curse of Correlation in Security Games

The issue illustrated in the previous motivating examples is due to the inherent correlation among the protection status of different targets when allocating a limited number of resources. The coverage of some targets must imply that some other targets are unprotected. These examples show how the attacker can take advantage of such correlation and infer a significant amount of information about the protection of other targets by monitoring even a single target. This is what we term the *Curse of Correlation* (CoC) in security games. The following Proposition tries to capture this phenomenon in a more formal sense.

We begin with some notation. Recall that any mixed strategy $\mathbf{p}$ is a distribution over the set $\mathcal{E}$ of pure strategies. Equivalently, we can view a mixed strategy as a random binary vector $\mathbf{X} = (X_1, \ldots, X_n) \in \{0,1\}^n$ satisfying $\mathbf{Pr}(X = \mathbf{e}) = p_{\mathbf{e}}$. Here, $X_i \in \{0,1\}$ denotes the random protection status of target i, and $\mathbf{Pr}(X_i = 1) = x_i$ is the marginal coverage probability. For any X_i, let $H(X_i) = x_i \log x_i + (1 - x_i)\log(1 - x_i)$ denote its Shannon entropy. Note that, the X_i's are correlated. Let $X_i|X_k$ denote X_i conditioned on X_k and $H(X_i|X_k)$ denote its conditional entropy. We say that target k is *trivial* if $\mathbf{Pr}(X_k = 1) = 0$ or 1; otherwise, k is *non-trivial*. Obviously, the attacker infers no information about other targets by monitoring a trivial target. The following proposition shows that if any non-trivial target k is monitored, the attacker can always infer information about the protection of other targets.[1]

Proposition 10. *For any non-trivial target k, we have*

$$\mathbf{E}_{X_k}\big[\sum_{i\neq k} H(X_i|X_k)\big] < \sum_{i\neq k} H(X_i)$$

.

Proof. Let $\mathbf{Pr}(X_i = 1|X_k = 1) = x_i^1$ and $\mathbf{Pr}(X_i = 1|X_k = 0) = x_i^0$. Since $\mathbf{E}_{X_k}[\mathbf{Pr}(X_i = 1|X_k)] = \mathbf{Pr}(X_i = 1) = x_i$, we have $x_i = x_i^1 \cdot \mathbf{Pr}(X_k = 1) + x_i^0 \cdot \mathbf{Pr}(X_k = 0)$. Since $H(X_i)$ is strictly concave w.r.t. x_i, we have

$$\mathbf{E}_{X_k} H(X_i|X_k) \geq H(X_i).$$

Summing over all $i \neq k$, we get $\sum_{i\in[n]:i\neq k} \mathbf{E}_{X_k} H(X_i|X_k) \geq \sum_{i\in[n]:i\neq k} H(X_i)$.

We now argue that the "=" *cannot* hold, and will prove it by contradiction. Note that target i is non-trivial; therefore $\mathbf{Pr}(X_k) \neq 0, 1$. Since $H(X_i)$'s are strictly concave, if the "=" holds, then we must have $\mathbf{Pr}(X_i = 1|X_k = 1) = \mathbf{Pr}(X_i = 1|X_k = 0) = x_i$ for any $i \neq k$. However, this implies that target i is trivial, i.e., i is either fully protected or unprotected. Otherwise, there

[1]Here we assume that all resources are fully used, and do not consider the (unreasonable) situations in which certain security resources are sometimes underused or idle.

must exist some $j \neq i$ such that the marginal probability of j will be different between the circumstances that i is protected and not protected. □

Proposition 10 shows that, conditioned on X_k, the entropy sum of all other X_i's strictly decreases in expectation. Note that it holds regardless of whether the security resources have scheduling constraints or not. This illustrates that the correlations among targets are intrinsic and inevitable.

Chapter 8

The Algorithmic Foundation for Dealing with Information Leakage

Most security games assume that the attacker only knows the defender's mixed strategy, but is not able to observe (even partially) the instantiated pure strategy. This fails to capture the cases where the attacker conducts real-time surveillance and may get partial observation regarding the deployed pure strategy. Despite its potential presence in reality as illustrated in Chapter 7, such issues, which we refer to as *information leakage*, have not been payed much attention in the literature on security games.

In this chapter, we propose two natural models of security games with information leakage, depending on how much the defender knows about the leakage situation. We then undertake an algorithmic study for the problem of computing the optimal defender strategy under leakage, and focus on perhaps the most basic setting: zero-sum security games with no scheduling constraints. We first describe an exponential-size LP formulation to compute the defender's optimal strategy against leakage, and then exhibit evidence of computational intractability for the model. This shows the intrinsic difficulty of handling leakage. We then tackle the problem from two different angles: developing polynomial-time algorithms for restricted settings and designing algorithms with approximation guarantees.

8.1 Information Leakage in Security Games – Two Basic Models

To the best of our knowledge, there has not been any previous study about security games with information leakage. Therefore, for simplicity, we start with a basic model where information leaks from *only one* target, though our model and algorithms can be generalized. For our algorithmic analysis in this chapter, we will focus on the simple security game setting where the defender allocates k resources to protect n targets without any scheduling constraint. Such models have applications in real security systems like ARMOR for the LAX airport and GUARDS for port patrolling (Tambe, 2011).

Consider a standard zero-sum Stackelberg security game with a defender and an attacker. The defender allocates k security resources to protect n targets, denoted by the set $[n] = \{1, 2, \ldots, n\}$. In this section we focus on the case where the security resources do *not* have scheduling constraints. As a result, any subset of $[n]$ with cardinality at most k is a defender pure strategy. For any $i \in [n]$, let r_i be the reward [c_i be the cost] of the defender when the attacked target i is protected [unprotected]. Since the game is zero-sum, the attacker's utility is the negation of the defender's utility. Following the notation in Section 2.2, we still use $\mathbf{e} \in \{0, 1\}^n$ to denote a pure strategy and $\mathcal{E}$ to denote the set of all pure strategies. Recall that we may also view $\mathbf{e}$ as a *subset* of $[n]$, denoting the protected targets. The intended interpretation should be clear from context. The *support* of a mixed strategy is the set of pure strategies with non-zero probabilities. Without information leakage, the problem of computing the defender's optimal mixed strategy can be compactly formulated as linear program (8.1) with each variable x_i as the marginal probability of covering target i. Any feasible marginal vector $\vec{x}$ can be efficiently implemented as a distribution over pure strategies, e.g., by Comb Sampling (Tsai, Yin, young Kwak, Kempe, Kiekintveld, & Tambe, 2010).

$$\begin{array}{lll} \text{maximize} & u & \\ \text{subject to} & u \le r_i x_i + c_i(1 - x_i), & \text{for } i \in [n]. \\ & \sum_{i \in [n]} x_i \le k & \\ & 0 \le x_i \le 1, & \text{for } i \in [n]. \end{array} \tag{8.1}$$

Building on this basic security game, our model goes one step further and considers the possibility that the protection status of one target leaks to the attacker. Here, by "protection status" we mean whether this target is protected or not in an *instantiation* of the mixed strategy. We consider two basic models of information leakage.

8.1.1 Adversarial Leakage

In the **AD**versarial **I**nformation **L**eakage (ADIL) model, parameterized by a probability parameter $p_0 \in [0, 1]$, we assume that with probability $1 - p_0$, one *adversarially* chosen target leaks information, and otherwise no target leaks information. Our goal then is to compute the optimal defender strategy assuming such an adversarially chosen leaking target. This model captures the case where the attacker will strategically choose a target for surveillance and with a certain probability he succeeds in observing the protection status of the surveiled target. In practice, the parameter p_0 can be estimated by domain experts. This model is suitable for the situation where the defender does not know much about the leakage situation except knowing that some target may leak information. The model then takes a robust perspective and optimizes against the worst case.

8.1.2 Probabilistic Leakage

In the **PR**obabilistic **I**nformation **I**eakage (PRIL) model, parameterized by probabilities $p_i(\geq 0)$ for $i = 0, 1, \ldots, n$, we assume that the leaking target is i with probability p_i, and with probability p_0 no targets leak information. Therefore, these probabilities satisfy $p_0 + \sum_{i=1}^{n} p_i = 1$, i.e., $\vec{p} = (p_0, p_1, \ldots, p_n) \in \Delta_{n+1}$ where Δ_{n+1} is the $(n+1)$-dimensional simplex. In practice, the vector $\vec{p}$ is usually given by domain experts and may be determined by the nature or property of targets. This model requires the defender to have much more knowledge about the game. It is suitable when the defender has a relatively good estimate of the leakage situation and such an estimate is summarized as a distribution over the leakage probabilities of targets.

8.2 Complexity Barriers to Computing the Optimal Strategy

Given either leakage model – PRIL parameterized by $\vec{p} \in \Delta_{n+1}$ or ADIL parameterized by p_0– we are interested in computing the optimal defender mixed strategy. Recall that we focus on zero-sum security games.

To see what an optimal strategy is like under a particular leakage model and what kind of information we need to keep track of, we start with a simple illustrative example. Consider a zero-sum security game with 4 targets and 2 resources. The profile of rewards r_i [cost c_i] is $\vec{r} = (1, 1, 2, 2)$ [$\vec{c} = (-2, -2, -1, -1)$], where the coordinates are indexed by target ids. If there is no information leakage, it is easy to see that the optimal marginal coverage is $\vec{x} = (\frac{2}{3}, \frac{2}{3}, \frac{1}{3}, \frac{1}{3})$. The attacker will attack an arbitrary target, resulting in a defender utility of 0. Now, let us consider a simple case of information leakage. Assume the attacker observes whether target 1 is protected or not in any instantiation of the mixed strategy, i.e., $p_1 = 1$. As we will argue, how the marginal probability $\vec{x}$ is implemented would matter now. One way to implement $\vec{x}$ is to protect targets $\{1, 2\}$ with probability $\frac{2}{3}$ and protect $\{3, 4\}$ with probability $\frac{1}{3}$. However, this implementation is "fragile" in the presence of the above information leakage. In particular, if the attacker observes that target 1 is protected (which occurs with probability $\frac{2}{3}$), he infers that the defender is protecting targets $\{1, 2\}$ and will attack 3 or 4, resulting in a defender utility of -1; if target 1 is not protected, the attacker will just attack, resulting in a defender utility of -2. Therefore, the defender gets expected utility $-\frac{4}{3}$.

Now consider another way to implement the *same* marginal $\vec{x}$ by the following mixed strategy:

$\{1,2\}$	$\{1,3\}$	$\{1,4\}$	$\{2,3\}$	$\{2,4\}$	$\{3,4\}$
$10/27$	$4/27$	$4/27$	$4/27$	$4/27$	$1/27$

If the attacker observes that target 1 is protected (which occurs with probability $\frac{2}{3}$), then he infers that target 2 is protected with probability $\frac{\frac{10}{27}}{\frac{10}{27}+\frac{4}{27}+\frac{4}{27}} = \frac{5}{9}$, and target $3, 4$ are both protected with probability $\frac{2}{9}$. Some calculation shows that the attacker will have the same utility $\frac{1}{3}$ on targets $2, 3, 4$ and thus will choose an arbitrary one to attack, resulting in a defender utility of $-\frac{1}{3}$. On the other hand, if target 1 is observed to be unprotected, the defender gets utility -2. In expectation, the defender gets utility $\frac{2}{3} \times (-\frac{1}{3}) + \frac{1}{3} \times (-2) = -\frac{8}{9}$.

As seen above, though implementing the same marginals, the latter mixed strategy achieves better defender utility than the former one in the presence of information leakage. However, is it optimal? It turns out that the following mixed strategy achieves an even better defender utility of $-\frac{1}{3}$, which can be proved to be optimal: protect $\{1, 2\}$ with probability $\frac{5}{9}$, $\{1, 3\}$ with probability $\frac{2}{9}$ and $\{1, 4\}$ with probability $\frac{2}{9}$.

This example shows that compact representation by marginal coverage probabilities is not sufficient for computing the optimal defending strategy assuming information leakage. This naturally raises new computational challenges: how can we formulate the defender's optimization problem and compute the optimal solution? Is there still a compact formulation or is it necessary to enumerate all the exponentially many pure strategies? What is the computational complexity of this problem? These are the questions we aim to answer in this section.

8.2.1 An Exponential-Size LP Formulation and Evidence of Hardness

We will focus on the PRIL model. The formulation for the ADIL model will be provided at the end of this section since it admits a similar derivation. Fixing the defender's mixed strategy, let T_i ($\neg T_i$) denote the event that target i is *protected* (*unprotected*). For the PRIL model, the defender's utility equals

$$DefU = p_0 u + \sum_{i=1}^{n} p_i (u_i + v_i)$$

where $u = \min_j [r_j \mathbf{Pr}(T_j) + c_j \mathbf{Pr}(\neg T_j)]$ is the defender's utility when there is no information leakage; and

$$\begin{aligned} u_i &= \mathbf{Pr}(T_i) \times \min_j [r_j \mathbf{Pr}(T_j|T_i) + c_j \mathbf{Pr}(\neg T_j|T_i)] \\ &= \min_j [r_j \mathbf{Pr}(T_j, T_i) + c_j \mathbf{Pr}(\neg T_j, T_i)] \end{aligned}$$

is the defender's utility when target i leaks out its protection status as T_i (i.e., protected) multiplied by probability $\mathbf{Pr}(T_i)$. Similarly

$$v_i = \min_j [r_j \mathbf{Pr}(T_j, \neg T_i) + c_j \mathbf{Pr}(\neg T_j, \neg T_i)]$$

is the defender's expected utility multiplied by probability $\mathbf{Pr}(\neg T_i)$ when target i leaks status $\neg T_i$ (i.e., unprotected).

Define variables $x_{ij} = \mathbf{Pr}(T_i, T_j)$ (setting $x_{ii} = \mathbf{Pr}(T_i)$). Using the fact that $\mathbf{Pr}(T_i, \neg T_j) = x_{ii} - x_{ij}$ and $\mathbf{Pr}(\neg T_i, \neg T_j) = 1 - x_{ii} - x_{jj} + x_{ij}$, we obtain the following linear program which computes the defender's optimal patrolling strategy:

$$\begin{array}{lll} \text{maximize} & p_0 u + \sum_{i=1}^{n} p_i(u_i + v_i) & \\ \text{subject to} & u \le r_j x_{jj} + c_j(1 - x_{jj}), & \text{for } j \in [n]. \\ & u_i \le r_j x_{ij} + c_j(x_{ii} - x_{ij}), & \text{for } i, j \in [n]. \\ & v_i \le r_j(x_{jj} - x_{ij}) + c_j(1 - x_{ii} - x_{jj} + x_{ij}), & \text{for } i, j \in [n]. \\ & x_{ij} = \sum_{\mathbf{e}: i,j \in \mathbf{e}} \theta_e, & \text{for } i, j \in [n]. \\ & \sum_{\mathbf{e} \in \mathcal{E}} \theta_e = 1 & \\ & \theta_e \ge 0, & \text{for } \mathbf{e} \in \mathcal{E}. \end{array} \tag{8.2}$$

where $u, u_i, v_i, x_{ij}, \theta_e$ are variables; $\mathbf{e}$ denotes a pure strategy and the sum condition "$\mathbf{e} : i, j \in \mathbf{e}$" means summing over all the pure strategies that protect both targets i and j (or i if $i = j$); θ_e denotes the probability of choosing strategy $\mathbf{e}$.

Unfortunately, LP (8.2) suffers from an exponential explosion of variables, specifically, θ_e. From the sake of computational efficiency, one natural idea is to find a compact representation of the defender's mixed strategy. As suggested by LP (8.2), the variables x_{ij}, indicating the probability that targets i, j are both protected, are sufficient to describe the defender's objective and the attacker's incentive constraints.

Let us call the variables x_{ij} the *pairwise marginals* and think of them as a matrix $X \in \mathbb{R}^{n \times n}$, i.e., the i'th row and j'th column of X is x_{ij} (not to be confused with the *marginals* $\vec{x}$). We say X is *feasible* if there exists a mixed strategy, i.e., a distribution over pure strategies, that achieves the pair-wise marginals X. Clearly, not all $X \in \mathbb{R}^{n \times n}$ are feasible. Let $\mathcal{P}(n, k) \subseteq \mathbb{R}^{n \times n}$ be the set of all *feasible* X. The following lemma shows a structural property of $\mathcal{P}(n, k)$.

Lemma 15. *$\mathcal{P}(n, k)$ is a polytope and any $X \in \mathcal{P}(n, k)$ is a* symmetric positive semi-definite *(PSD) matrix.*

Proof. Notice that X is feasible if and only if there exists θ_e for any pure strategy $\mathbf{e}$ such that the following linear constraints hold:

$$\begin{array}{ll} x_{ij} = \sum_{\mathbf{e}: i,j \in \mathbf{e}} \theta_e, & \text{for } i, j \in [n]. \\ \sum_{\mathbf{e} \in \mathcal{E}} \theta_e = 1 & \\ \theta_e \ge 0, & \text{for } \mathbf{e} \in \mathcal{E}. \end{array} \tag{8.3}$$

These constraints define a polytope for variables $(X, \vec{\theta})$. Therefore, its projection to the lower dimension X, which is precisely $\mathcal{P}(n, k)$, is also a polytope.

To prove $X \in \mathcal{P}(n, k)$ is PSD, we first observe that any vertex of $\mathcal{P}(n, k)$, characterizing a pure strategy, is PSD. In fact, let $\mathbf{e} \in \{0, 1\}^n$ be any pure strategy. Then the pair-wise marginal

w.r.t. $\mathbf{e}$ is $X_e = \mathbf{e} \cdot \mathbf{e}^T$, which is PSD. Therefore, any $X \in \mathcal{P}$, which is a convex combination of its vertices, is also PSD. □

$$\begin{array}{lll} \text{maximize} & p_0 u + \sum_{i=1}^{n} p_i(u_i + v_i) & \\ \text{subject to} & u \le r_j x_{jj} + c_j(1 - x_{jj}), & \text{for } j \in [n]. \\ & u_i \le r_j x_{ij} + c_j(x_{ii} - x_{ij}), & \text{for } i, j \in [n]. \\ & v_i \le r_j(x_{jj} - x_{ij}) + c_j(1 - x_{ii} - x_{jj} + x_{ij}), & \text{for } i, j \in [n]. \\ & X \in \mathcal{P}(n, k) & \end{array} \tag{8.4}$$

With Lemma 15, we may re-write LP (8.2) compactly as LP (8.4) with variables u, u_i, v_i and X. Therefore, we would be able to compute the optimal strategy in polynomial time if there are only polynomially many constraints determining the polytope $\mathcal{P}(n, k)$ — recall that this is the approach we took with LP (8.1) in the case of no information leakage. Unfortunately, the following lemma rules out the approach of using compact representations of polytopes (unless P = NP).

Lemma 16. *Optimizing over $\mathcal{P}(n, k)$ is NP-hard.*

Proof. We prove the lemma by reduction from the densest k-subgraph problem. Given any graph instance $G = (V, E)$, let A be the adjacency matrix of G. Consider the following linear program:

$$\begin{array}{ll} \text{maximize} & \sum_{i,j \in [n]} A_{ij} x_{ij} \\ \text{subject to} & X \in \mathcal{P}(n, k). \end{array} \tag{8.5}$$

This linear program must have a *vertex* optimal solution X^* which satisfies $X^* = \mathbf{e}\mathbf{e}^T$ for some pure strategy $\mathbf{e} \in \{0, 1\}^n$. Therefore, the linear objective satisfies

$$\sum_{i,j \in [n]} A_{ij} x_{ij} = tr(AX^*) = tr(A \times \mathbf{e}\mathbf{e}^T) = tr(\mathbf{e}^T A \mathbf{e}) = \mathbf{e}^T A \mathbf{e}.$$

Notice that $\mathbf{e}^T A \mathbf{e}/2k$ equals the density of a subgraph of G with k nodes indicated by $\mathbf{e}$. Since X^* is the optimal solution to LP (8.5), it also maximizes the density $\mathbf{e}^T A \mathbf{e}/2k$ over all subgraphs with k nodes. In other words, the ability to optimize LP (8.5) implies the ability to compute the densest k-subgraph, which is NP-hard. Therefore, optimizing over $\mathcal{P}(n, k)$ is NP-hard. □

Lemma 16 suggests that there is no hope of finding polynomially many linear constraints which determine $\mathcal{P}(n, k)$ or, more generally, an efficient separation oracle for $\mathcal{P}(n, k)$, assuming P≠NP. In fact, $\mathcal{P}(n, k)$ is closely related to a fundamental geometric object, known as the *correlation polytope*, which has applications in quantum mechanics, statistics, machine learning and combinatorial problems. The following is a formal definition of the correlation polytope.

Definition 4. *(Pitowsky, 1991) Given an integer n, the* Correlation Polytope $\mathcal{P}(n)$ *is defined as follows*

$$\mathcal{P}(n) = Conv\left(\{vv^T : v \in \{0,1\}^n\}\right).$$

where $Conv(S)$ denotes the convex hull of set S. Notice that $vv^T \in \{0,1\}^{n\times n}$ for all $v \in \{0,1\}^n$.

The following proposition shows an interesting connection between $\mathcal{P}(n,k)$ and the correlation polytope $\mathcal{P}(n)$.

Proposition 11. *$X \in \mathcal{P}(n,k)$ if and only if the following three constraints hold: (a) $X \in \mathcal{P}(n)$; (b) $tr(X) = \sum_{i=1}^n x_{ii} = k$; and (c) $sum(X) = \sum_{i,j=1}^n x_{ij} = k^2$. In other words, $\mathcal{P}(n,k)$ is decided by $\mathcal{P}(n)$ with two additional linear constraints.*

Proof. We show that, given $X \in \mathcal{P}(n)$, if X satisfies the following two linear constraints: $tr(X) = k,\ sum(X) = k^2$, then $X \in \mathcal{P}(n,k)$.

Since $X \in \mathcal{P}(n)$, there exist $X_i \in \mathcal{P}(n,i)$ and $p_i \geq 0$, such that $X = \sum_{i=1}^n p_i X_i$ and $\sum_{i=1}^n p_i = 1$. That is, X is a convex combination of elements from each $\mathcal{P}(n,i)$. Notice that $\forall X_i \in \mathcal{P}(n,i)$, we have $tr(X_i) = i$ and $sum(X_i) = i^2$, since any vertex of $\mathcal{P}(n,i)$ satisfies these constraints. Let $X \in \mathcal{P}(n,k)$. Then we have:

$$(i) :1 = \sum_{i=1}^n p_i$$

$$(ii) :k = tr(X) = \sum_{i=1}^n p_i \times tr(X_i) = \sum_{i=1}^n p_i \times i$$

$$(iii) :k^2 = sum(X) = \sum_{i=1}^n p_i \times sum(X_i) = \sum_{i=1}^n p_i \times i^2$$

By the Cauchy-Schwarz inequality, we have $(\sum_{i=1}^n p_i)(\sum_{i=1}^n p_i i^2) \geq (\sum_{i=1}^n p_i i)^2$. Plugging the above three equations into the Cauchy-Schwarz inequality yields that the equality holds. The condition of equality for the Cauchy-Schwarz inequality is that $p_i i^2/p_i$ is a constant for all i, such that $p_i \neq 0$. This shows that there is only one non-zero element among the p_i's. That is $p_k = 1$. Therefore, $X \in \mathcal{P}(n,k)$. □

We note that (Pitowsky, 1991) defines correlation polytopes in a more general fashion, and our definition of $\mathcal{P}(n)$ is in fact an important special case of the correlation polytope, which is called the full correlation polytope. (Pitowsky, 1991) proved that checking for membership of polytope $\mathcal{P}(n)$ is NP-complete.

Remark. Though Lemma 16 does not directly imply the NP-hardness of computing the optimal defender strategy (i.e., solving LP (8.4)), it serves as strong evidence. In (Xu, 2016), it was proved

that for standard security games with no information leakage, the NP-hardness of optimizing over the polytople of marginal probabilities implies the NP-hardness of computing the optimal defender strategy. Thus we view Lemma 16 as an indicator of the difficulty for computing the optimal defender strategy, though we remark that whether such an implication holds rigorously in the setting with information leakage is an interesting open problem.

8.2.2 The Dual Program and Evidence of Hardness

Another popular approach for computing the optimal defender strategy in security games is to use the technique of column generation, which is a master/slave decomposition of an optimization problem (Tambe, 2011; Jain, Korzhyk, Vaněk, Conitzer, Pěchouček, & Tambe, 2011). The essential part of this approach is the slave problem (Jain et al., 2010). Next we show that this approach will not work either. In particular, we will first derive the slave problem and then show that it is NP-hard to solve.

A slave problem is an important subproblem for solving security games with a large number of pure strategies using the *Column Generation* technique. Any algorithm for solving the slave problem is also called a "defender oracle" by convention (Jain et al., 2010). We now derive the formulation for the slave problem in the PRIL model.

Recall that LP (8.2) has a large number of variables because the number of pure strategies is exponential. However, by counting the number of activated constraints at optimality, we know that only polynomially many of these pure strategies will have non-zero probabilities at optimality since most pure strategies activate the corresponding constraint $\theta_e \geq 0$ and take probability 0. Column generation is based on this observation, i.e., the optimal mixed strategy has small support. Basically, instead of solving LP (8.2) on the set $\mathcal{E}$ of all pure strategies, it starts from a small subset of pure strategies, denoted as $\mathcal{A}$, and solves the following "restricted" LP.

$$
\begin{array}{lll}
\text{maximize} & p_0 u + \sum_{i=1}^{n} p_i (u_i + v_i) & \\
\text{subject to} & u \leq r_j x_{jj} + c_j (1 - x_{jj}), & \text{for } j \in [n]. \\
& u_i \leq r_j x_{ij} + c_j (x_{ii} - x_{ij}), & \text{for } i, j \in [n]. \\
& v_i \leq r_j (x_{jj} - x_{ij}) + c_j (1 - x_{ii} - x_{jj} + x_{ij}), & \text{for } i, j \in [n]. \\
& x_{ij} = \sum_{\mathbf{e} \in \mathcal{A}: i,j \in \mathbf{e}} \theta_e, & \text{for } i, j \in [n]. \\
& \sum_{\mathbf{e} \in \mathcal{A}} \theta_e = 1 & \\
& \theta_e \geq 0, & \text{for } \mathbf{e} \in \mathcal{A}.
\end{array}
\tag{8.6}
$$

Notice that the only difference between LP (8.2) and LP (8.6) is that the set $\mathcal{E}$ of all pure strategies is replaced by a small subset $\mathcal{A}$. In practice, $\mathcal{A}$ is usually initialized with a small number of pure strategies that are arbitrarily chosen. Column generation proceeds roughly as follows: 1. it solves LP (8.6); 2. by checking the dual of LP (8.6) the defender oracle judges whether the computed

optimal solution for LP (8.6) is also optimal for LP (8.2) (assigning all pure strategies in $\mathcal{E} \setminus \mathcal{A}$ probability 0); if not, the oracle finds a new pure strategy to be added to the set $\mathcal{A}$ and updates $\mathcal{A}$. This procedure continues until the defender oracle asserts that the computed optimal solution w.r.t. the current $\mathcal{A}$ is also optimal for LP (8.2). We now explain the underlying rationale of the column generation technique.

We first derive the dual of LP (8.6). In fact, to emphasize the key aspects and avoid messy derivations, we rewrite LP (8.6) in the following abstract form:

$$\begin{array}{lll} \text{maximize} & d^T y & \\ \text{subject to} & Mx + Ny \leq c & \\ & x_{ij} - \sum_{\mathbf{e} \in \mathcal{A}: i,j \in \mathbf{e}} \theta_e = 0, & \text{for } i,j \in [n]. \\ & \sum_{\mathbf{e} \in \mathcal{A}} \theta_e = 1 & \\ & \theta_e \geq 0, & \text{for } \mathbf{e} \in \mathcal{A}. \end{array} \tag{8.7}$$

where the variable y represents the vector consisting of u, v_i, u_i while the variable x is the vector representation of x_{ij} (putting i, j in some fixed order); d is a vector summarizing the original objective coefficients; the constraints $Mx + Ny \leq c$ summarize the first three sets of constraints in LP (8.6). This abstract form not only simplifies our derivation of the dual; more importantly it emphasizes that the column generation technique works regardless of what the first three sets of constraints are as long as there are polynomially many of them.

Let $M_{index(i,j)}$ be the *column vector* of M corresponding to variable x_{ij} and N_l be the *column vector* of N corresponding to the l'th component of y. We can now simply derive the dual of LP (8.7) as follows:

$$\begin{array}{lll} \text{minimize} & c^T \rho + \omega & \\ \text{subject to} & \rho^T N_l \geq d_l, & \text{for all } l. \\ & \rho^T M_{index(i,j)} + \beta_{ij \geq 0}, & \text{for } i,j \in [n]. \\ & -\sum_{i,j \in \mathbf{e}} \beta_{ij} + \omega \geq 0, & \text{for } \mathbf{e} \in \mathcal{A}. \\ & \rho \geq 0 & \end{array} \tag{8.8}$$

where ρ are the dual variables w.r.t. the first set of constraints in LP (8.7) and β_{ij}, ω are the dual variables w.r.t. the second and third set of constraints.

Note that the optimal solution to LP (8.7) (denoted as $OptSol_{\mathcal{A}}$) and the optimal solution to LP (8.8) (denoted as $OptSolDual_{\mathcal{A}}$) can both be computed efficiently when A is small. A key observation here is that, if $OptSolDual_{\mathcal{A}}$, in particular, ω and β_{ij}, happens to make the constraints $-\sum_{i,j \in \mathbf{e}} \beta_{ij} + \omega \geq 0$, $\forall \mathbf{e} \in \mathcal{A}$ hold more generally as $-\sum_{i,j \in \mathbf{e}} \beta_{ij} + \omega \geq 0$, $\forall \mathbf{e} \in \mathcal{E}$, then we claim that $OptSol_{\mathcal{A}}$ is also an optimal solution to LP (8.2) (by picking pure strategies in $\mathcal{E} \setminus \mathcal{A}$ with probability 0). This is because, if we replace $\mathcal{A}$ by $\mathcal{E}$ in both LP (8.7) and LP (8.8), $OptSol_{\mathcal{A}}$ is still feasible to LP (8.7) because all the newly added strategies (in $\mathcal{E} \setminus \mathcal{A}$) have

probability 0; $OptSolDual_{\mathcal{A}}$ is still feasible to LP (8.8) because our ω, β_{ij} make constraints $-\sum_{i,j\in\mathbf{e}} \beta_{ij} + \omega \geq 0$ hold for all $\mathbf{e} \in \mathcal{E}$ by assumption. Furthermore, *complementary slackness* still holds since the added new variables in LP (8.7) all take value 0. By linear program basics, $OptSol_{\mathcal{A}}$ is still optimal if we replace $\mathcal{A}$ in LP (8.7) by $\mathcal{E}$, which is precisely LP (8.2).

As a result, our key task is to judge whether $-\sum_{i,j\in\mathbf{e}} \beta_{ij} + \omega \geq 0$ holds for all $\mathbf{e} \in \mathcal{E}$ for a given dual solution. This is equivalent to deciding whether $\omega \geq \max_{\mathbf{e}\in\mathcal{E}} \left[\sum_{i,j\in\mathbf{e}} \beta_{ij}\right]$. Therefore, the slave problem is defined as follows.

Slave Problem: *For any given weights β_{ij}, solve the following maximization problem:*

$$\max_{\mathbf{e}\in\mathcal{E}} \left[\sum_{i,j\in\mathbf{e}} \beta_{ij}\right] = \max_{\mathbf{e}\in\mathcal{E}} s^T (\frac{M + M^T}{2}) s \tag{8.9}$$

where M is the matrix satisfying $M_{ij} = \beta_{ij}$. In other words, the defender oracle finds a pure strategy $\mathbf{e}$ that maximizes the sum $\sum_{i,j\in\mathbf{e}} \beta_{ij}$.

Recall that any algorithm that solves the slave problem is called a defender oracle. With this oracle, column generation proceeds as follows: 1. compute LP (8.7) and LP (8.8); 2. use the defender oracle to solve Problem (8.9): if the optimal value is less than or equal to the dual variable ω, asserts optimality; otherwise, add $\mathbf{e}^*$ – the optimal solution to Problem (8.9) – to $\mathcal{A}$; 3. repeat until optimality is reached. Notice that the newly added $\mathbf{e}^*$ does not belong to the original $\mathcal{A}$ because all $\mathbf{e} \in \mathcal{A}$ satisfy $\sum_{i,j\in\mathbf{e}} \beta_{ij} \leq \omega$. Column generation does not guarantee polynomial convergence, but usually converges very fast in practice. This is because the optimal mixed strategy usually has small support.

The following lemma shows that the slave problem is also NP-hard, and thus rules out the efficient implementation of the column generation approach for solving the problem.

Lemma 17. *The slave problem described above is NP-hard.*

Proof. The proof is similar to that of Lemma 16 by viewing the matrix M as an adjacency matrix of a graph. We omit the repetition here. □

By now, we have exhibited evidence of the hardness for solving LP (8.2) using either compact representation or the technique of column generation. For the ADIL model, a similar derivation

yields that the following LP formulation computes the optimal defender strategy. It is easy to verify that it shares its marginal probabilities and slave problem with the PRIL model.

$$\begin{array}{lll} \text{maximize} & p_0 u + (1-p_0) w & \\ \text{subject to} & u \le r_j x_{jj} + c_j(1 - x_{jj}), & \text{for } j \in [n]. \\ & u_i \le r_j x_{ij} + c_j(x_{ii} - x_{ij}), & \text{for } i, j \in [n]. \\ & v_i \le r_j(x_{jj} - x_{ij}) + c_j(1 - x_{ii} - x_{jj} + x_{ij}), & \text{for } i, j \in [n]. \\ & w \le u_i + v_i, & \text{for } i \in [n]. \\ & X \in \mathcal{P}(n, k) & \end{array} \tag{8.10}$$

where variable w is the defender's expected utility when an adversarially chosen target is observed by the attacker. LP (8.10) can also be abstractly written in the form of LP (8.7), and thus its slave problem is also NP-hard.

8.3 Provable Algorithms for Restricted Settings and Approximate Solutions

The results in Section 8.2 suggest the difficulty of developing a polynomial-time algorithm to exactly solve security games with leakage. In this section, we seek to tackle this computational challenge by focusing on well-motivated special settings.

8.3.1 Leakage from Small Support

Despite the hardness results for the general case, we show that the slave problem admits a polynomial time algorithm if the information only possibly leaks from a small subset of targets; we call this set the *leakage support*. By reordering the targets, we may assume without loss of generality that only the first m targets, denoted by the set $[m]$, could possibly leak information in both the PRIL and ADIL model. For the PRIL model, this means $p_i = 0$ for any $i > m$ and for the ADIL model, this means the attacker only chooses a target in $[m]$ for surveillance.

Why does this make the problem tractable? Intuitively the reason is as follows: when information leaks from a small set of targets, we only need to consider the correlations between these leaking targets and others, which is a much smaller set of variables than in LP (8.2) or (8.10). When restricted to a leakage support of size m, a similar derivation as in Section 8.2.2 reveals that the slave problem is the follows. Let A be a *symmetric* matrix of the following block form

Slave Problem with Leakage Support $[m]$**:** *Let A be a symmetric matrix of the following block form*

$$A : \begin{bmatrix} A_{mm} & A_{mm'} \\ A_{m'm} & A_{m'm'} \end{bmatrix} \tag{8.11}$$

where $m' = n - m$; $A_{mm'} \in \mathbb{R}^{m \times m'}$ for any integers m, m' is a sub-matrix and, crucially, $A_{m'm'}$ is a diagonal matrix. Given A of the form (8.11)*, find a pure strategy* $\mathbf{e}$ *such that* $\mathbf{e}^T A \mathbf{e}$ *is maximized.*

A defender oracle will identify the size-k principal submatrix with maximum entry sum for any A of form (8.11). Note that $m = n$ in the general case. Next, we prove that the slave problem admits a polynomial time algorithm when m is a constant. We start with some notation. Let $A[i,:]$ be the i'th row of matrix A and $diag(A)$ be the vector consisting of the diagonal entries of A. For any subsets C_1, C_2 of $[n]$, let A_{C_1,C_2} be the submatrix of A consisting of rows in C_1 and columns in C_2, and $sum(A_{C_1,C_2}) = \sum_{i \in C_1, j \in C_2} A_{ij}$ be the entry sum of A_{C_1,C_2}. The following lemma shows that Algorithm 6 solves the slave problem. The key insight here is that for a pure strategy $\mathbf{e}$ to be optimal, once the set $C = \mathbf{e} \cap [m]$ is decided, its complement $\overline{C} = \mathbf{e} \setminus C$ can be explicitly identified. Therefore we can simply brute-force search to find the best $C \subseteq [m]$. Lemma 18 provides the algorithm's guarantee, which then yields the polynomial-time solvability for the case of small m (Theorem 8.3.1).

Lemma 18. *Let m be the size of the leakage support. Algorithm 6 solves the slave problem and runs in $poly(n, k, 2^m)$ time. In particular, the slave problem admits a $poly(n, k)$ time algorithm if m is a constant.*

Proof. First, it is easy to see that Algorithm 1 runs in $poly(2^m, n, k)$ time since the for-loop is executed at most 2^m times. We show that it solves the slave problem.

Let $\mathbf{e}$ denote the indices of the principal submatrix of A with maximum entry sum. Notice that $\mathbf{e}$ can also be viewed as a pure strategy. Let $C = \mathbf{e} \cap [m]$ and $\overline{C} = \mathbf{e} \setminus C$. We claim that, given C, $\overline{C}$ must be the set of indices of the largest $k - |C|$ values from the set $\{v_{m+1}, \ldots, v_n\}$, where $\vec{v}$ is defined as $\vec{v} = 2\sum_{i \in C} A[i,:] + diag(A)$. In other words, if we know C, the set $\overline{C}$ can be easily identified. To prove the claim, we re-write the $sum(A_{s,s})$ as follows:

$$
\begin{aligned}
& sum(A_{s,s}) \\
= \;& sum(A_{C,C}) + 2sum(A_{C,\overline{C}}) + sum(A_{\overline{C},\overline{C}}) \\
= \;& sum(A_{C,C}) + 2sum(A_{C,\overline{C}}) + sum(diag(A_{\overline{C},\overline{C}})) \\
= \;& sum(A_{C,C}) + sum(2\sum_{i \in C} A_{i,\overline{C}} + diag(A_{\overline{C},\overline{C}})) \\
= \;& sum(A_{C,C}) + sum(v_{\overline{C}}) \\
= \;& val_C
\end{aligned}
$$

where $\vec{v} = 2\sum_{i \in C} A[i,:] + diag(A)$ and $v_{\overline{C}}$ is the sub-vector of v with indices in $\overline{C}$. Given C, $sum(A_{C,C})$ is fixed; therefore $\overline{C}$ must be the set of indices of the largest $k - |C|$ elements from

$\{v_{m+1}, \ldots, v_n\}$. Algorithm 1 then loops over all the possible $C \subseteq [m]$ (2^m many) and identifies the optimal one, i.e., the one achieving the maximum val_C. □

Algorithm 6: Defender Oracle

Input: matrix A of the form (8.11).
Output: a pure strategy $\mathbf{e}$.

1: **for** all $C \subseteq [m]$ constrained by $|C| \leq k$ **do**
2: $\vec{v} = 2\sum_{i \in C} A[i,:] + diag(A)$;
3: Choose the largest $k - |C|$ values from the set $\{v_{m+1}, \ldots, v_n\}$, and denote the set of their indices as $\overline{C}$;
4: Set $val_C = sum(A_{C,C}) + sum(v_{\overline{C}})$;
5: **return** the pure strategy $\mathbf{e} = C \cup \overline{C}$ with maximum val_C.

Utilizing Lemma 18, we can prove the following theorem.

Theorem 8.3.1. *(***Polynomial Solvability***) There is a $poly(n, k)$ time algorithm which computes the optimal defender strategy in the PRIL and ADIL model, if the size of the leakage support m is a constant.*

Proof. We prove that LP (8.7) (which is really LP (8.4) written abstractly) can be solved in polynomial time. In fact, we prove that its dual program can be solved in polynomial time, which then implies that the primal LP (8.7) can be solved in polynomial time due to complementary slackness (Grötschel et al., 1988).

Since the leaking target could only be from $[m]$, LP (8.7) only has variables x_{ij} for any $i \in [m]$ or $j \in [m]$ or $i = j$. As a result, the dual LP (8.8) only has variable β_{ij}'s for $i \in [m]$ or $j \in [m]$ or $i = j$, which satisfies precisely the condition in the above slave problem for small support $[m]$. This implies that the polynomial time defender oracle (Algorithm 6) can be used to efficiently evaluate whether the constraints $-\sum_{i,j \in \mathbf{e}} \beta_{ij} + \omega \geq 0$ are violated or not. All other other (polynomially many) constraints can be explicitly evaluated. Therefore, an efficient defender oracle gives rise to an efficient separation oracle for the feasible region of the dual LP (8.8). As a result, we can solve the dual program in polynomial time, concluding the proof. □

8.3.2 An Approximation Algorithm

We now consider approximation algorithms. Recall that information leakage is due to the correlation between targets. Thus one natural way to minimize leakage is to allocate each resource *independently* with certain distributions. The normalized marginal $\vec{x}^*/k$ is a natural choice, where $\vec{x}^*$ is the solution to LP (8.1). To avoid the waste of using multiple resources to protect the same target, we sample without replacement. Formally, the *independent sampling without replacement*

(`IndepSamp`) algorithm proceeds as follows: 1. compute the optimal solution $\vec{x}^*$ of LP (8.1); 2. independently sample k elements from $[n]$ *without replacement* using the distribution $\vec{x}^*/k$.

Since players may have positive or negative utilities in zero-sum games, a multiplicative approximation ratio in terms of utility is not meaningful. To analyze the performance of this algorithm, we instead consider the "coverage-match" criterion — i.e., how many more security resources are needed in order to achieve the same coverage level as the case of no leakage? More formally, we say that an algorithm is an α-approximation ($\alpha \geq 1$) if the protection statuses $T_1, \ldots, T_n$ it induces satisfy that for any i, $\mathbf{Pr}(T_i) \geq x_i^*$, $\mathbf{Pr}(T_i|T_j) \geq x_i^*$ and $\mathbf{Pr}(T_i|\neg T_j) \geq x_i^*$ for any target j that possibly leaks information.[1] This guarantees that the marginal protection probability of any target i is at least x_i^*, i.e., i's protection probability in the SSE with no leakage, conditioned on any target j with possible leakage.

Theorem 8.3.2 shows that `IndepSamp` with a slight modification[2] is roughly a $(\frac{e}{e-1})$-approximation under the aforementioned coverage-match criterion for both the PRIL and ADIL models.

Theorem 8.3.2. *Under the coverage-match criterion, there is a $\frac{e}{e-\frac{k-1}{k-2}}$-approximation algorithm for both the PRIL and ADIL model.*

Proof of Theorem 8.3.2

Let $Y = Y(\vec{x}) \in \mathbb{R}^{n \times n}$ be a function of any $\vec{x} \in \mathbb{R}^n$, where y_{ij} is the probability that targets i, j are both protected using `IndepSamp`. Let T_i ($\neg T_i$) denote the event that target i is *protected* (*unprotected*) using `IndepSamp`. We first prove Lemma 19, which provides a lower bound regarding how well the pair-wise marginals in Y approximate the original marginals $\vec{x}$. The difficulty of proving Lemma 19 lies in the fact that Y does not have a closed form in terms of $\vec{x}$ if we sample without replacement. Our proof is based on a coupling argument by relating the algorithm to independent sampling *with* replacement.[3]

Lemma 19. *Given $\vec{x}$, $Y = Y(\vec{x})$ satisfies the following (in)equalities:*

$$\mathbf{Pr}(T_j) = y_{jj} \geq (1 - \frac{1}{e})x_j, \forall j \in [n]; \tag{8.12}$$

$$\mathbf{Pr}(T_j|T_i) = \frac{y_{ij}}{y_{ii}} \geq (\frac{k-2}{k-1} - \frac{1}{e})x_j, \forall i \neq j. \tag{8.13}$$

$$\mathbf{Pr}(T_j|\neg T_i) = \frac{y_{jj} - y_{ij}}{1 - y_{ii}} \geq (1 - \frac{1}{e})x_j, \forall i \neq j. \tag{8.14}$$

[1] Recall that T_i is the event that target i is protected.

[2] Because directly applying `IndepSamp` can never match, e.g., coverage probability 1.

[3] Our insistence on sampling without replacement is due to a practical consideration — making complete use of all security resources, though using the sampling approach with replacement may be easier to analyze from a theoretical perspective.

Proof. To prove these inequalities, we instead consider independent sampling *with* replacement. Define the function $Z = Z(\vec{x}) \in \mathbb{R}^{n \times n}$ to be a function of $\vec{x}$, where z_{ij} is the probability that targets i, j are protected together when sampling with replacement. Contrary to Y, Z has a succinct closed forms, and therefore we can lower bound entries in Z. We first consider z_{jj}.

$$\begin{aligned} z_{jj} &= 1 - (1 - x_j/k)^k \\ &\geq 1 - e^{-x_j} \\ &\geq (1 - \frac{1}{e})x_j. \end{aligned}$$

where we used the fact $(1-\epsilon)^{\frac{1}{\epsilon}} \leq e^{-1}$ for any $\epsilon \in (0,1)$. Now we lower bound z_{ij}/z_{ii} as follows.

$$\begin{aligned} \frac{z_{ij}}{z_{ii}} &= \frac{1 - (1 - \frac{x_i}{k})^k - (1 - \frac{x_j}{k})^k + (1 - \frac{x_i}{k} - \frac{x_j}{k})^k}{1 - (1 - x_i/k)^k} \\ &= 1 - (1 - \frac{x_j}{k})^k - \frac{(1 - \frac{x_i}{k})^k (1 - \frac{x_j}{k})^k - (1 - \frac{x_i}{k} - \frac{x_j}{k})^k}{1 - (1 - x_i/k)^k} \\ &= 1 - (1 - \frac{x_j}{k})^k - \frac{(1 - \frac{x_i}{k})^k}{1 - (1 - x_i/k)^k}[(1 - \frac{x_j}{k})^k - (1 - \frac{x_j}{k - x_i})^k] \\ &\geq (1 - \frac{1}{e})x_j - \frac{e^{-x_i}}{1 - e^{-x_i}}[(1 - \frac{x_j}{k})^k - (1 - \frac{x_j}{k - x_i})^k] \end{aligned} \tag{8.15}$$

where all the equations use arithmetic, while the inequality uses the fact that $(1 - \frac{x_j}{k})^k \leq e^{-x_j}$ and $-\frac{x}{1-x}$ is a decreasing function of $x \in (0,1)$. We now upper-bound the term $(1 - \frac{x_j}{k})^k - (1 - \frac{x_j}{k-1})^k$ using the formula $a^k - b^k = (a-b)\sum_{i=0}^{k-1} a^i b^{k-1-i}$, as follows

$$\begin{aligned} & (1 - \frac{x_j}{k})^k - (1 - \frac{x_j}{k - x_i})^k \\ = \; & (1 - \frac{x_j}{k} - 1 + \frac{x_j}{k - x_i}) \sum_{t=0}^{k-1} (1 - \frac{x_j}{k})^t (1 - \frac{x_j}{k - x_i})^{k-1-t} \\ \leq \; & \frac{x_j x_i}{k(k - x_i)} \times k \\ \leq \; & \frac{x_i x_j}{k - 1} \end{aligned}$$

Plugging the above upper bound back into Inequality 8.15, we thus have

$$\begin{aligned} \frac{z_{ij}}{z_{ii}} &\geq (1 - \frac{1}{e})x_j - \frac{e^{-x_i}}{1 - e^{-x_i}} \frac{x_i x_j}{k - 1} \\ &= (1 - \frac{1}{e})x_j - \frac{x_i}{e^{x_i} - 1} \frac{x_j}{k - 1} \\ &\geq (1 - \frac{1}{e})x_j - \frac{x_j}{k - 1} \\ &= (\frac{k - 2}{k - 1} - \frac{1}{e})x_j, \end{aligned}$$

where the last inequality is due to the fact that $f(x) = \frac{x}{e^x - 1}$ is a decreasing function for $x \in (0, 1)$ and is upper bounded by $\lim_{x \to 0} \frac{x}{e^x - 1} = 1$.

Finally, we have

$$\begin{aligned}
\frac{z_{jj} - z_{ij}}{1 - z_{ii}} &= \frac{(1 - \frac{x_i}{k})^k - (1 - \frac{x_i}{k} - \frac{x_j}{k})^k}{(1 - \frac{x_i}{k})^k} \\
&= 1 - (1 - \frac{x_j}{k - x_i})^k \\
&\geq 1 - (1 - x_j/k)^k \\
&\geq (1 - \frac{1}{e})x_j.
\end{aligned}$$

To prove the lemma, we only need to show that $y_{jj} \geq z_{jj}$, $y_{ij}/y_{ii} \geq z_{ij}/z_{ii}$ and $(y_{jj} - y_{ij})/(1 - y_{ii}) \geq (z_{jj} - z_{ij})/(1 - z_{ii})$. To prove these inequalities, we use a coupling argument. Consider the following two stochastic process (StoP):

1. $StoP^1$: at time t independently sample a random value i_t ($\in [n]$) with probability x_{i_t}/k for any $t = 1, 2, \ldots$ until precisely k *different* elements from $[n]$ show up.

2. $StoP^2$: at time t independently sample a random value i_t ($\in [n]$) with probability x_{i_t}/k for $t = 1, 2, \ldots k$.

Let C^1 [C^2] denote all the possible random sequences generated by $StoP^1$ [$StoP^2$], and C^1_j [C^2_j] denote the subset of C^1 [C^2], which consists of all the sequences including at least one j. For any $e \in C^2_j$, let C_e be the subset of sequences in C^1, whose first k elements are precisely e. Notice that any sequence in C^1 has length at least k while any sequence in C^2 has precisely k elements. Furthermore, $C_e \subseteq C^1_j$ and $C_e \cap C_{e'} = \emptyset$ for any $e, e' \in C^2_j$ and $e \neq e'$.

Now, think of each sequence as a probabilistic event generated by the stochastic process. Notice that $P(e; StoP^2) = P(C_e; StoP^1)$ due to the independence of the sampling procedure. Therefore, we have

$$\begin{aligned}
P(C^2_j; StoP^2) &= \sum_{e \in C^2_j} P(e; StoP^2) \\
&= \sum_{e \in C^2_j} P(C_e; StoP^1) \\
&\leq P(C^1_j; StoP^1)
\end{aligned}$$

However, $P(C^1_j | StoP^1) = y_{jj}$ and $P(C^2_j | StoP^2) = z_{jj}$. This proves $y_{jj} \geq z_{jj}$.

Notice that $y_{ij}/y_{ii} \geq z_{ij}/z_{ii}$ is equivalent to $P(e \in C^2_j | e \in C^2_i; StoP^2) \geq P(e \in C^1_j | e \in C^1_i; StoP^1)$. To prove this inequality, we claim that it is without loss of generality to assume the first sample is i in both processes. This is because, if the first i shows up as the t'th sample, moving i to the first position would not change the probability of the sequence due to independence

between the sampling steps. Conditioned on i being sampled first, a similar argument as above shows that the probability of Stochastic process $StoP^1$ generating j is at least the probability of stochastic process $StoP^2$ generating j.

Finally, $(y_{jj} - y_{ij})/(1 - y_{ii}) \geq (z_{jj} - z_{ij})/(1 - z_{ii})$ is equivalent to $P(e \in C_j^2 | e \notin C_i^2; StoP^2) \geq P(e \in C_j^1 | e \notin C_i^1; StoP^1)$. The conditional probability $P(e \in C_j^2 | e \notin C_i^2; StoP^2)$ can be viewed as the probability of generating a sequence including element j in a modified $StoP^2$ — it generates any $j \neq i$ with probability $x_j^*/(k - x_i^*)$ but generates i with probability 0. Viewing from this perspective, we can conclude $P(e \in C_j^2 | e \notin C_i^2; StoP^2) \geq P(e \in C_j^1 | e \notin C_i^1; StoP^1)$ using a similar argument for proving $y_{jj} \geq z_{jj}$. □

Let $\vec{x}^*$ be the optimal solution to LP (8.1) and let $\alpha = \frac{e}{e - \frac{k-1}{k-2}}$. One natural idea is to scale up $\vec{x}^*$ by a factor of α and then apply `IndepSamp`. The problem here is that some targets may have probability larger than 1 after the scaling up. To deal with this issue, we divide targets into two sets: $S = \{j : x_j^* < 1/\alpha\}$ and $S^{\complement} = [n] \setminus S = \{j : x_j^* \geq 1/\alpha\}$. For any $j \in S^{\complement}$, we simply cover it with probability 1. Note that these targets will satisfy $\mathbf{Pr}(T_j) \geq x_j^*$ and will not leak any information about the protection of other targets since they will be always protected. For targets in S, we scale up their marginal probability by a factor of α and then apply the `IndepSamp` algorithm. In total we need no more than αk resources. By Lemma 19, we know that $\mathbf{Pr}(T_j) \geq x_j^*$, $\mathbf{Pr}(T_j | T_i) \geq x_j^*$ and $\mathbf{Pr}(T_j | \neg T_i) \geq x_j^*$ for any $j \neq i \in S$.

To summarize, for any target $i \in [n]$ that possibly leaks its protection status (either protected or unprotected), the conditional protection probability of any other target j is always at least x_j^*. Therefore, in the ADIL leakage model, regardless which target i leaks information, the conditional protection probability of any other target j is always at least x_j^*. This also holds in the PRIL model.

Chapter 9

Mitigating Harms of Information Leakage via Entropy Maximization

Chapter 8 proposed and studied the complexity of two basic leakage models. Unfortunately, even in simple security game settings, we easily encounter barriers of computational intractability. Therefore, to obtain solutions with rigorous guarantees, we have to restrict ourselves to even more specific settings as described in Section 8.2. In this chapter, we instead propose a heuristic approach, based on max-entropy sampling, for handling information leakage. The solutions in Chapter 8 only work for the setting with no scheduling constraints. However, the framework we describe in this chapter will be generalizable to security games with arbitrary scheduling constraints. Together with some other practical advantages (illustrated later), this makes the approach very appealing in various real-world security applications.

9.1 The Max-Entropy Sampling Framework

9.1.1 Max-Entropy Sampling Over General Set Systems

As we mentioned in Section 2.2.1, the set of defender pure strategies can be viewed as a set system, or equivalently, a set of binary vectors. Classic security games seek to achieve certain marginal probability vector $\vec{x}$ (indexed by targets) by randomizing over these binary vectors. From Carathéodory's theorem we know that, given any marginal vector $\vec{x}$ in the convex hull of $\mathcal{E}$, denoted as $\mathbf{conv}(\mathcal{E})$, there are usually many different mixed strategies that achieve the same $\vec{x}$ (e.g., see examples in Section 7.1). One question then is which of these mixed strategies is more robust to information leakage. One natural choice is the mixed strategy of maximum entropy subject to achieving the given marginal vector $\mathbf{x}$. Intuitively, this is because the max-entropy distribution is the most unpredictable distribution and usually has low correlation among targets.

In this section, we describe a general framework for computing the max-entropy distribution over the set system $\mathcal{E}$ subject to matching any given marginal $\vec{x} \in \mathbf{conv}(\mathcal{E})$. This problem has

been studied in the literature of theoretic computer science; see, e.g., (Jerrum, Valiant, & Vazirani, 1986; Singh & Vishnoi, 2013). Our description here serves more as a review of previous work or its variants.

Computing the max-entropy distribution can be formulated as the solution to an $\mathcal{O}(2^n)$-size Convex Program (CP (9.1)) where variable θ_e is the probability of taking pure strategy $\mathbf{e}$.

$$\begin{array}{lll} \text{maximize} & \sum_{\mathbf{e}\in\mathcal{E}} -\theta_e \ln(\theta_e) & \\ \text{subject to} & \sum_{\mathbf{e}:\, i\in\mathbf{e}} \theta_e = x_i, & \text{for } i \in [n]. \\ & \sum_{\mathbf{e}\in\mathcal{E}} \theta_e = 1 & \\ & \theta_e \geq 0, & \text{for } \mathbf{e} \in \mathcal{E}. \end{array} \tag{9.1}$$

Convex program for computing the max-entropy distribution

An obvious challenge for solving CP (9.1) is that the optimal θ^* typically has exponentially large support, and thus cannot even be written down explicitly in polynomial time. This can be overcome via algorithms that efficiently sample a set $\mathbf{e}$ "on the fly". Therefore, we say that an algorithm solves CP (9.1) if it takes $\mathbf{x}$ as input and randomly samples $\mathbf{e} \in \mathcal{E}$ with probability θ^*_e where θ^* is the optimal solution to CP (9.1).

Sampling the max-entropy distribution is closely related to the following *generalized counting* problem over $\mathcal{E}$.

Definition 5 (Generalized Counting)**.** *Given any $\alpha \in \mathbb{R}^n_+$, compute $C(\alpha) = \sum_{\mathbf{e}\in\mathcal{E}} \alpha_e$, where $\alpha_e = \prod_{i\in\mathbf{e}} \alpha_i$.*

Observe that $C(\mathbf{1})$ equals precisely the cardinality of $\mathcal{E}$. More generally, $C(\alpha)$ is a weighted count of the elements in $\mathcal{E}$ with weights $\alpha_e = \prod_{i\in\mathbf{e}} \alpha_i$. The relation between max-entropy sampling and counting is through the following *unconstrained and convex* dual program of CP (9.1) with variables $\vec{\beta} \in \mathbb{R}^n$ and $e^{-\beta_e} = \Pi_{i\in\mathbf{e}} e^{-\beta_i}$.

$$\text{minimize} \quad f(\vec{\beta}) = \sum_{i=1}^n \beta_i x_i + \ln(\sum_{\mathbf{e}\in\mathcal{E}} e^{-\beta_e}), \tag{9.2}$$

Dual program of the convex program (9.1).

The following theorem characterizes the optimal solutions for CP (9.1) and will be useful for our later results.

Theorem 9.1.1. *(Singh & Vishnoi, 2013) Let $\vec{\beta}^* \in \mathbb{R}^n$ be the optimal solution to CP* (9.2) *and set $\alpha_i = e^{-\beta^*_i}$ for any $i \in [n]$. Then, the optimal solution of CP* (9.1) *satisfies*

$$\theta^*_e = \frac{\alpha_e}{\sum_{\mathbf{e}'\in\mathcal{E}} \alpha_{e'}}, \tag{9.3}$$

where $\alpha_e = \Pi_{i \in \mathbf{e}} \alpha_i$ *for any pure strategy* $\mathbf{e} \in \mathcal{E}$.

Furthermore, if the generalized counting problem over $\mathcal{E}$ *can be solved in* $\mathrm{poly}(n)$ *time, then* $\vec{\beta}^*$ *can be computed in* $poly(n)$ *time.*

Proof. The characterization of θ_e^* is based on the KKT conditions of CP (9.1) and its dual program (9.2). Its proof can be found in (Singh & Vishnoi, 2013); we thus will not repeat the argument. Here, we prove the prescriptive part of the theorem. In particular, we will show that $\vec{\beta}^*$ can be computed in $poly(n)$ time given any polynomial-time algorithm for the generalized counting problem and moreover, our algorithm will be practically efficient as well.

Notice that CP (9.2) has n variables but an expression of exponentially many terms, in particular, the sum $\sum_{\mathbf{e} \in \mathcal{E}} e^{-\beta_e}$. The essential difficulty in evaluating $f(\vec{\beta})$ lies in computing the sum $\sum_{\mathbf{e} \in \mathcal{E}} e^{-\beta_e}$, since the other parts can be explicitly calculated in $\mathrm{poly}(n)$ time. Note that calculating $\sum_{\mathbf{e} \in \mathcal{E}} e^{-\beta_e}$ is precisely the generalized counting problem over the set system $\mathcal{E}$ with weights $\alpha_i = e^{-\beta_i}$ for $i \in [n]$. As a result, if we have a $\mathrm{poly}(n)$ time counting oracle, we can evaluate the function value of $f(\vec{\beta})$ in $\mathrm{poly}(n)$ time. With this $\mathrm{poly}(n)$ time value oracle, one can conclude that CP (9.1) can be solved in $\mathrm{poly}(n)$ time using the ellipsoid method (Grötschel et al., 1988), though the order of this polynomial is usually large.

Here, we instead give a more practical algorithm. We show that the gradient can also be evaluated efficiently. Therefore, one can use standard gradient-descent based algorithm to solve CP (9.1) which is usually more efficient in practice. In particular,

$$\frac{\partial f(\vec{\beta})}{\partial \beta_i} \quad = \quad x_i - \frac{\sum_{\mathbf{e} \in \mathcal{E}: i \in \mathbf{e}} e^{-\beta_e}}{\sum_{\mathbf{e} \in \mathcal{E}} e^{-\beta_e}}.$$

The only non-trivial part of evaluating $\frac{\partial f(\vec{\beta})}{\partial \beta_i}$ is to compute $\sum_{\mathbf{e} \in \mathcal{E}: i \in \mathbf{e}} e^{-\beta_e}$. This can be calculated by employing a generalized counting algorithm twice: once for the weights $e^{-\beta_j}$ for each $j \in [n]$ and once with the same weights except using $2e^{-\beta_i}$ for i. Their difference equals precisely $\sum_{\mathbf{e} \in \mathcal{E}: i \in \mathbf{e}} e^{-\beta_e}$.

To sum up, given a $\mathrm{poly}(n)$ time algorithm for the generalized counting problem, we can evaluate $f(\vec{\beta})$ and its gradient in $\mathrm{poly}(n)$ time. Thus we can also optimize the function in $\mathrm{poly}(n)$ time.

□

After computing the optimal dual solution $\vec{\beta^*}$, we need to develop sampling algorithms that output strategy $\mathbf{e}$ with probability precisely $\theta_e^* = \frac{\alpha_e}{\sum_{\mathbf{e}' \in \mathcal{E}} \alpha_{e'}}$. This process will depend on the setting. However, it can usually be done efficiently given a generalized counting algorithm, which is the case in all the settings we study.

9.1.2 Why Maximizing Entropy?

As we mentioned before, the issue of information leakage arises due to the correlation among the protection statues of targets, a phenomenon which we term the *curse of correlation* (CoC) in Section 7.3. To deal with CoC, the ideal approach is to come up with an accurate model to capture the attacker's partial observation, i.e., an information *leakage model*, and then solve the model to obtain the defender's optimal defending strategy, as we did in Section 8.2. However, we note that this approaches suffers from several drawbacks.

1. **Unavailability of an Accurate Leakage Model.** The attacker's choice of target monitoring depends on many hidden factors, and thus is highly unpredictable. Therefore, it is typically very difficult to know which targets are leaking information — otherwise the defender could have resolved the issue in the first place via other approaches. As a result, it is usually not possible to obtain an accurate leakage model. As we will illustrates in our experiments, optimizing over an inaccurate leakage model can even be harmful to the defender compared to doing nothing.

2. **Scalability and Computational Barriers.** Even if the defender has an accurate leakage model, computing the optimal defender strategy against the leakage model is intractable generally. As we mentioned in Section 8.2, even in the simplest possible model — zero-sum games, no scheduling constraints and a single target leaking information — we exhibit evidence of intractability. The problem becomes even more difficult in more complicated spatio-temporal settings with scheduling constraints, e.g., the motivating examples in Chapter 7.

3. **Vulnerability to Attacker's Strategic Manipulations.** Another concern about any optimal solution tailored to a specific leakage model is that such a solution may be easily "gamed" by the attacker. In particular, the optimal solution naturally biases towards the leaking targets by assigning more security forces to these targets. This, however, opens the door for the attacker to strategically manipulate the defender's belief on leaking targets, e.g., by intentionally spreading misleading information, with the goal of shifting the defense away from the attacker's prime targets. As we show in our experiments, this could cause significant loss to the defender.

Entropy maximization — a more robust solution. These barriers motivate our adoption of the more robust (though inevitably more conservative) max-entropy approach, as illustrated in Section 9.1.1. We propose to first compute the optimal defender strategy assuming no leakage and then play the mixed strategy with maximum entropy subject to matching the desired marginal

probabilities. Our choice of max entropy is due to at least three reasons. First, the max-entropy strategy is the most random, and thus unpredictable, defender strategy. When the defender is uncertain about which target is leaking information (the setting we are in), we believe that taking the most random strategy is one natural choice. Second, the max-entropy distribution usually exhibits substantial approximate stochastic independence among the protection statuses of targets[1], so that the protection status of any leaking target does not carry much information about that of others. Third, as we will illustrate in the next few sections, the max entropy approach performs well in comparisons with several other alternatives in simulated games; in fact, in some settings, it achieves a solution quality that is even close to the optimal defender utility under no leakage! Given such encouraging empirical results, we believe that entropy maximization stood out as a powerful approach to address information leakage.

From a practical perspective, the max-entropy approach also enjoys several advantages. First, it does not require a concrete leakage model. Instead, it seeks to reduce the overall correlation among the statuses of all targets, and thus serves as a robust solution. Second, this approach is easily "compatible" with any current deployed security systems since it does not require any change to previously deployed algorithms while only adding randomness (in some sense, this is a *strictly* better solution than previous ones). This is particularly useful in domains where re-building a new security system is not feasible or too costly.

9.2 Security Settings with No Scheduling Constraints

As an instantiation of the above framework, we first consider the simple security game setting with no scheduling constraints. In this case, a defender pure strategy is any subset of $[n]$ of size k. Such models have applications in real security systems like ARMOR for LAX airport and GUARDS for port patrolling in general (Tambe, 2011).

9.2.1 A Polynomial-Time Max-Entropy Sampling Algorithm

In this section, we prove the following theorem.

Theorem 9.2.1. *When there are no scheduling constraints, the distribution that maximizes entropy subject to matching any given marginal* $\mathbf{x} \in \mathbf{conv}(\mathcal{E})$ *can be sampled in* $\mathrm{poly}(n)$ *time.*

The proof of Theorem 9.2.1 relies on the following two lemmas.

Lemma 20. *When there are no scheduling constraints, the generalized counting problem over* $\mathcal{E}$*for any given weight admits a* $\mathrm{poly}(n)$ *time algorithm.*[2]

[1] This is widely observed in practice, and also theoretically proved in some settings, e.g., matchings (Kahn & Kayll, 1997).

[2] *The set system* $\mathcal{E}$ *is also known as the uniform matroid in this case.*

Proof. Let $\vec{\alpha} = (\alpha_1, \ldots, \alpha_n)$ be any given weight vector. Our goal is to compute the sum $\sum_{\mathbf{e}\in\mathcal{E}} \alpha_e$ where $\alpha_e = \Pi_{i\in\mathbf{e}}\alpha_i$.

We show that a dynamic program computes the sum $\sum_{\mathbf{e}\in\mathcal{E}} \alpha_e$ in $poly(n)$ time. Note that the set of all pure strategies consists of all the subsets of $[n]$ of cardinality k. We build the following DP table $T(i,j) = \sum_{\mathbf{e}:\mathbf{e}\subseteq[j],|\mathbf{e}|=i} \alpha_e$, which sums over all the subsets of $[j]$ of cardinality i. Our goal is to compute $T(k,n) = \sum_{\mathbf{e}\in\mathcal{E}} e^{-\beta_e}$. We first initialize $T(0,j) = 1$ and $T(1,j) = \Pi_{i=1}^{j}\alpha_i$ for any j. Then using the following update rule, we can build the DP table and compute $T(k,n)$ in $poly(k,n)$ time.

$$T(i,j) \quad = \quad T(i,j-1) + \alpha_j T(i-1,j-1).$$

This update rule is correct because $T(i,j)$ is the sum of two parts. The first part contains terms without element j. Therefore, these terms must sum up to $T(i,j-1)$. The second part contains terms with element j and other $i-1$ elements before j. These terms must sum up to $\alpha_j T(i-1,j-1)$. □

Lemma 20, together with Theorem 9.1.1, shows that we can solve CP (9.2) in $\mathrm{poly}(n)$ time. Our next lemma shows how to efficiently sample a pure strategy $\mathbf{e}$ from an exponentially large support with probability θ_e^* defined by Equation (9.3). The algorithm (Algorithm 7) simply goes through each target and adds it to the pure strategy with a specifically designed probability until exactly k targets are added.

Lemma 21. *Given any input $\vec{\alpha} \in [0,\infty)^n$, Algorithm 7 runs in $\mathrm{poly}(n)$ time and correctly samples a pure strategy $\mathbf{e}$ with probability $\theta_e = \frac{\alpha_e}{\sum_{\mathbf{e}\in\mathcal{E}}\alpha_e}$, where $\alpha_e = \Pi_{i\in\mathbf{e}}\alpha_i$.*

Proof. Note that Table $T(i,j)$ can be computed in $\mathrm{poly}(n)$. We first show that the "while" loop in Algorithm 7 terminates within at most n steps. In fact, j decreases by 1 each step and furthermore $j \geq i \geq 0$ always holds. This is because when j decreases until $j = i$, j will be sampled with probability $\frac{\alpha_j T(i-1,j-1)}{T(i,j)} = \frac{\alpha_i T(i-1,i-1)}{T(i,i)} = 1$; then both j and i will decrease by 1 (Steps $6-9$). This continues until $i = 0$. Furthermore, the algorithm terminates with $|\mathbf{e}| = k$ because the cardinality of $\mathbf{e}$ always satisfies $|\mathbf{e}| = k - i$ by Steps $6-8$ until the termination at $i = 0$. Therefore, Algorithm 7 runs in $\mathrm{poly}(n)$ time.

Now we show that Algorithm 2 outputs $\mathbf{e}$ with probability θ_e. Let the output $\mathbf{e} = \{i_1, \ldots, i_k\}$ be sorted in decreasing order, i.e., $i_1 > i_2 > \ldots > i_k$. Notice that

$$T(i,j) = \alpha_j T(i-1,j-1) + T(i,j-1).$$

Therefore, in the *Sampling* step (Step 5) of Algorithm 7, j is not included in $\mathbf{e}$ with probability $T(i,j-1)/T(i,j)$. Therefore, to sample $\mathbf{e} = \{i_1, \ldots, i_k\}$, it must be the case that $n, n-$

Algorithm 7: Max-entropy sampling in settings with no scheduling constraints

Input: : $\vec{\alpha} \in [0, \infty)^n$, k.

Output: : a pure strategy $\mathbf{e}$ with $|\mathbf{e}| = k$.

1: Initialize: $\mathbf{e} = \emptyset$; the DP table $T(0, j) = 1$ and $T(j, j) = \Pi_{i=1}^{j} \alpha_i$ for any $j \in [n]$.

2: Compute $T(i, j) = \sum_{\mathbf{e}:\mathbf{e}\subseteq[j],|\mathbf{e}|=i} \alpha_e$ for any i, j satisfying $i \leq k, j \leq n$ and $1 \leq i \leq j$, using the following update rule

$$T(i, j) \quad = \quad T(i, j-1) + \alpha_j T(i-1, j-1).$$

3: Set $i = k$, $j = n$;

4: **while** $i > 0$ **do**

5: Sampling: independently add j to $\mathbf{e}$ with probability

$$p_j = \frac{\alpha_j T(i-1, j-1)}{T(i, j)};$$

6: **if** j was added to $\mathbf{e}$ **then**

7: $i = i - 1$;

8: $j = j - 1$;

9: **return** $\mathbf{e}$.

$1, \ldots, i_1 + 1$ are not included, while i_1 is included; $i_1 - 1, \ldots, i_2 + 1$ are not included, while i_2 is included; and so on. In addition, the sampling in each of these steps is conditioned on all its previous steps and the probability of each step is known. Therefore, by multiplying these probabilities together, we have

$$\begin{aligned} P(s) \quad &= \quad \frac{T(k, n-1)}{T(k, n)} \times \frac{T(k, n-2)}{T(k, n-1)} \cdots \times \frac{\alpha_{i_1} T(k-1, i_1 - 1)}{T(k, i_1)} \\ &\quad \times \frac{T(k-1, i_1 - 2)}{T(k-1, i_1 - 1)} \cdots \frac{\alpha_{i_k} T(0, i_k - 1)}{T(1, i_k)} \\ &= \quad \frac{\Pi_{t \leq k} \alpha_{i_t}}{T(k, n)} \\ &= \quad \theta_e. \end{aligned}$$

This gives precisely the probability we want. □

9.2.2 A Linear-Time Heuristic Sampling Algorithm

Though the sampling algorithm in Section 9.2.1 provably runs in polynomial time, the order of the polynomial may be large due to repeated calls to the counting oracle. This limits the scalability of the algorithm in very large applications. In this section, we develop a linear-time heuristic sampling algorithm, termed *Uniform Comb Sampling* (UniCS). UniCS is extremely efficient and, as we will show, also performs well in practice.

(Tsai et al., 2010) presented the Comb Sampling algorithm, which randomly samples a pure strategy and achieves a given marginal in expectation. The algorithm can be elegantly described as follows (also see Figure 9.1): thinking of k resources as k buckets with height 1 each, we then put each target, the height of which equals precisely its marginal probability, one by one into the buckets. If one bucket gets full when filling in a certain target, we move the "rest" of that target to a new empty bucket. Continue this until all the targets are filled in, at which time we know that all k buckets are full. The algorithm then takes a horizontal line with a uniformly randomly chosen height from the interval $[0, 1]$, and the k targets intersecting the horizontal line constitute the sampled pure strategy. As easily observed, Comb Sampling achieves the marginal coverage in expectation (Tsai et al., 2010).

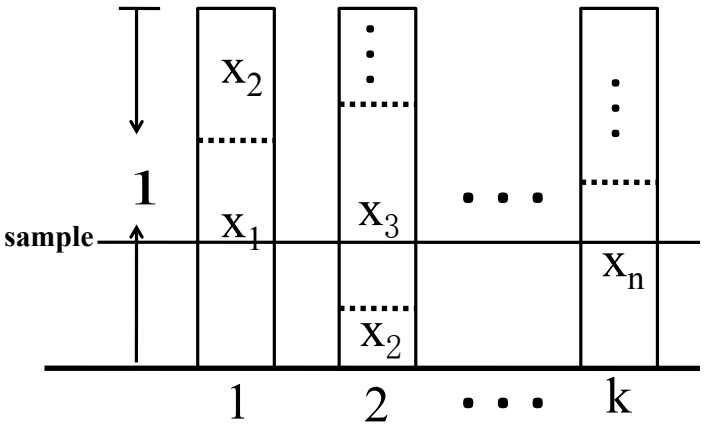


Figure 9.1: Comb sampling

However, is Comb Sampling robust against information leakage? We first observe that Comb Sampling generates a mixed strategy with support size at most $n + 1$, which precisely matches the upper bound of Carathéodory's theorem.

Proposition 12. *Comb Sampling generates a mixed strategy which mixes over at most $n+1$ pure strategies.*

Proof. In Figure 9.1, let the sample line move from height 0 to height 1 continuously. The sampled pure strategy changes only when it meets a dotted line in any bucket. There are at most $n - 1$ dotted lines (because there are n targets), so the total number of possible pure strategies is $(n - 1) + 2 = n + 1$. □

Proposition 12 suggests that the mixed strategy sampled by Comb Sampling might be very easy to explore. Therefore we propose a variant of the Comb Sampling algorithm. Our key

observation is that Comb Sampling achieves the marginal coverage regardless of the order of the targets. That is, the marginal is still obtained if we randomly shuffle the order of the targets *each time* before sampling, and then fill them in one by one. Therefore, we propose the following Uniform Comb Sampling (UniCS) algorithm:

1. Choose an order of the n targets uniformly at random;
2. Fill the targets into the buckets based on the random order, and then apply Comb Sampling.

This algorithm runs in linear time because: (1) a random permutation can be generated in linear time, e.g., using the Knuth Shuffle (Knuth, 1997); (2) the Comb Sampling algorithm runs in linear time. The property of UniCS is summarized in the following proposition.

Proposition 13. *Uniform Comb Sampling (UniCS) runs in $\mathcal{O}(n)$ time and achieves the marginal coverage probability.*

9.2.3 Experiments

In this section, we experimentally study how traditional algorithms and our new algorithms perform in the presence of *probabilistic* or *adversarial* information leakage (i.e., the PRIL and ADIL model in Section 8.1). Since we also have an algorithm that computes the exact optimal solution in this setting (Section 8.3) (though it runs in exponential time), we will also compare our max-entropy sampling (heuristic) approach with the exact optimal solution. In particular, we compare the following five algorithms.

- *Traditional*: optimal marginal + comb sampling, the traditional way to solve security games with no scheduling constraints (Kiekintveld et al., 2009; Tsai et al., 2010);
- *OPT*: the optimal algorithm for the PRIL or ADIL model (Section 8.1) using column generation with the defender oracle in Algorithm 6;
- *indepSample*: independent sampling without replacement (Section 8.3);
- *MaxEntro*: max entropy sampling (Algorithm 7);
- *UniCS*: uniform comb sampling.

All algorithms are tested on the following two sets of data:

Los Angeles International Airport (LAX) Checkpoint Data from (Pita et al., 2008b). This problem was modeled as a Bayesian Stackelberg game with multiple adversary types in (Pita et al., 2008b). To be consistent with our model, we instead only consider the game against one particular type of adversary — the terrorist-type adversary, which is the main concern of the

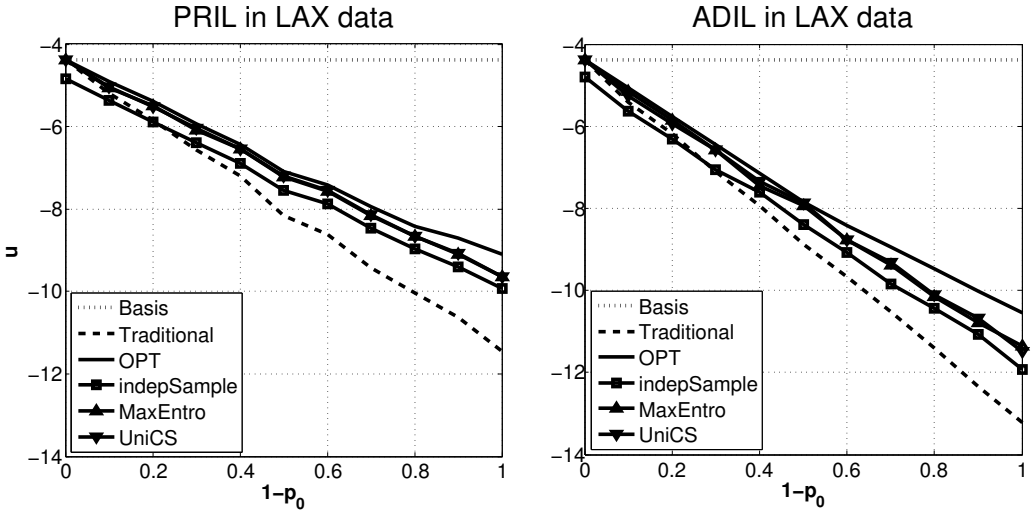


Figure 9.2: Comparisons on real LAX airport data.

airport. The defender's rewards and costs are obtained from (Pita et al., 2008b) and the game is assumed to be zero-sum in our experiments.

Simulated Game Payoffs. A systematic examination is conducted with simulated zero-sum security games with no scheduling constraints, i.e., the basic setting we studied in Section 8.1.[3] All generated games have 20 targets and 10 resources. The reward r_i (cost c_i) of each target i is chosen uniformly at random from the interval $[0, 10]$ ($[-10, 0]$). This corresponds to the covariant random game generator (Nudelman et al., 2004), with covariance equal to -1.

In terms of running time, all the algorithms run efficiently as expected (terminate within seconds using MATLAB) except the optimal algorithm *OPT*, which takes about 3 minutes per simulated game on average. Therefore we mainly compare defender utilities. All the comparisons are listed in Figure 9.2 (for LAX data) and Figure 9.3 (for simulated data). The line "Basis" is the utility with no leakage and is listed as a basis for utility comparisons. The Y-axis is the defender's utility — the higher, the better. We examine the effect of the *total probability of leakage* (i.e., the x-axis $1 - p_0$) on the defender's utility and consider $1 - p_0 = 0, 0.1, \ldots, 1$. For probabilistic information leakage, we randomly generate the probabilities that each target leaks information with the constraint $\sum_{i=1}^{n} p_i = 1 - p_0$. For the case of leakage from small support (for simulated payoffs only), we randomly choose a support of size 5. All the utilities are *averaged*

[3] Another rationale of focusing on zero-sum games is the following. Zero-sum games are strictly competitive; therefore, any information leaking to the attacker will benefit the attacker and hurt the defender. The effects of the curse of correlation (CoC) could be a mix of both good and bad aspects in general-sum security games because "leaking" information to the attacker there could sometimes be beneficial to the defender. This has been studied in Part I of these thesis on strategic information revelation in security games (Rabinovich et al., 2015; Guo et al., 2017). In zero-sum security games, however, any information to the attacker will hurt the defender. In this sense, zero-sum games serve as the best fit for studying harms of CoC. Previous work studying information leakage in normal-form games (Alon et al., 2013) also focused on zero-sum games.

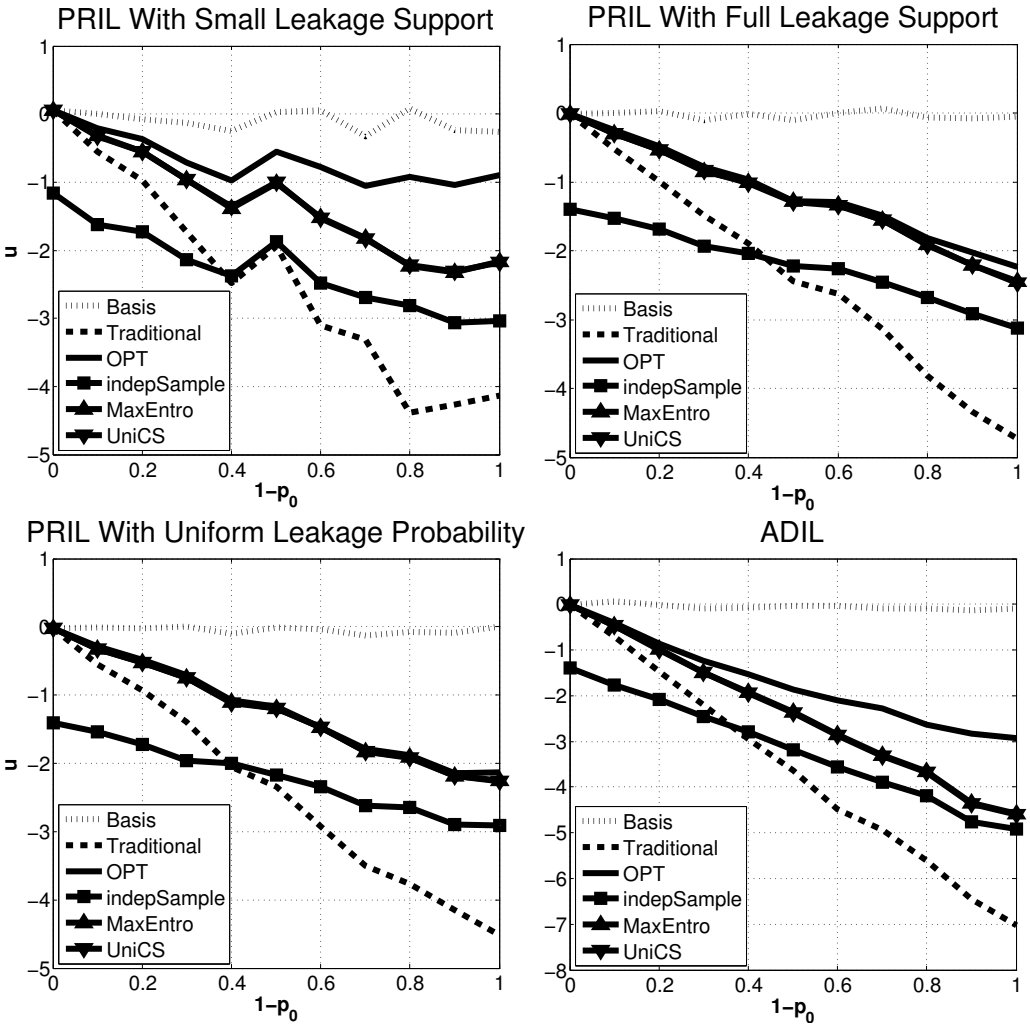


Figure 9.3: Comparisons in Simulated Games.

over 50 random games except the ADIL model for LAX data. For the simulated payoffs, we also consider a special case of uniform leakage probability of each target. The following observations follow from the figures.

Observation 1. The gap between the line "*Basis*" and "*OPT*" shows that information leakage from even one target may cause a dramatic utility decrease to the defender. Moreover, adversarial leakage causes more utility loss than probabilistic leakage; leakage from a restricted small support of targets causes less utility decrease than from full support.

Observation 2. The gap between the line "*OPT*" and "*Traditional*" demonstrates the necessity of handling information leakage. The relative loss $u(OPT) - u(Basis)$ is approximately half of the relative loss $u(Traditional) - u(Basis)$ in Figure 9.3 (and 65% in Figure 9.2). Furthermore, if leakage is from a small support (top-left panel in Figure 9.3), *OPT* is close to *Basis*.

Observation 3. *MaxEntro* and *UniCS* have almost the same performance (overlapping in all these figures). Both algorithms are almost optimal when the leakage support is the full set $[n]$ (they almost overlap with *OPT* in the top-right and bottom-left panels in Figure 9.3).

Observation 4. An interesting observation is that in all of these figures, *IndepSample* start to outperform *Traditional* roughly at $1 - p_0 = 0.3$ or 0.4, which is around $\frac{1}{e} \approx 0.37$. Furthermore, the gap between *IndepSample* and *OPT* does not change much at different $1 - p_0$.

Observation 5. From a practical view, if the leakage is from a small support, *OPT* is preferred as it admits efficient algorithms (Section 8.3); if the leakage is from a large support, *MaxEntropy* and *UniCS* are preferred as they can be computed efficiently and are close to optimality. From a theoretical perspective, we note that the intriguing performance of *IndepSample*, *MaxEntropy* and *UniCS* raises questions for future work.

9.3 The Air Marshal Scheduling Problem

In this section, we consider the problem of randomized air marshal scheduling, as illustrated in Section 7.1. One important task faced by the Federal Air Marshal Service (FAMS) is to schedule air marshals to protect *international* flights. In this setting, the schedule of each air marshal is a *round trip* (Kiekintveld et al., 2009), which is what we focus on.

We start by formally defining the problem. FAMS seeks to allocate k homogeneous air marshals to protect round-trip international flights originating from domestic cities to different outside cities. These round-trip flights constitute a bipartite graph $G = (A \cup B, E)$ in which nodes in A [B] correspond to all outbound [return] flights; $e = (A_i, B_j) \in E$ iff e forms a *consistent* round trip. We remind the reader that here we abuse notation since $\mathbf{e}$ is used to denote a pure strategy and $\mathcal{E}$ is the set of all defender pure strategies. Figure 9.4 depicts the graph between one domestic city and two outside cities, though in general we consider multiple domestic cities and multiple outside cities. Note that G is a union of multiple isolated smaller bipartite graphs, each containing all flights between two cities. This is because any flight from city a to city b can never form a round trip with a flight from city c to city a. We will call each isolated bipartite graph a *component*. Naturally, not any two flights A_i, B_j can form consistent round-trip flights. The following are natural constraints on the structure of the graph G: (A_i, B_j) forms a compatible round trip (i.e., $(A_i, B_j) \in E$) if (1) the destination city of A_i is the departure city of B_j; (2) the arrival time of A_i, denoted as $arr(A_i)$, and the departure time of B_j, denoted as $dep(Bj)$, satisfy $dep(B_j) - arr(A_i) \in [T_1, T_2]$ for constants $T_2 > T1 > 0$. Moreover, we assume that in any pure strategy, each flight is covered by at most one air marshal. This is a requirement that comes from the US Transportation Security Administration to ensure maximum usage of valuable security resources (Jain et al., 2010).

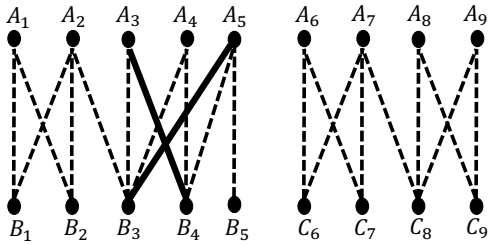


Figure 9.4: Consistent round-trip flights between a domestic city and two outside cities.

Next, we will develop a provably polynomial-time algorithm for sampling the max-entropy distribution as well as a fast heuristic sampling algorithm. Our exact algorithm crucially exploits certain "order" structure of the air marshal's schedules. The heuristic sampling algorithm can be generalized to other security games as well so long as we can efficiently compute the defender's best response. We evaluate these algorithms experimentally at the end of this section.

9.3.1 A Polynomial-Time Max-Entropy Sampling Algorithm

We prove the following theorem in this section.

Theorem 9.3.1. *In the federal air marshal scheduling problem with round trips, the distribution that maximizes entropy subject to matching any given marginal* $\mathbf{x} \in \mathbf{conv}(\mathcal{E})$ *can be sampled in* $\mathrm{poly}(n, k)$ *time, where* k *is the number of air marshals and* $n = |A \cup B|$ *is the number of total flights.*

The proof of Theorem 9.3.1 has two steps:

- First, we design a $\mathrm{poly}(n, k)$ time algorithm for the generalized counting problem over the set system $\mathcal{E}$ of defender pure strategies. By Theorem 9.1.1, this implies that we can compute the optimal solution to CP (9.2) in $\mathrm{poly}(n, k)$ time.
- Second, we will design an efficient sampling algorithm that samples a pure strategy $\mathbf{e}$ from an exponentially large support with probability θ^*_e as defined by Equation (9.3).

Step 1

We start with the first task. Let $G = (A \cup B, E)$ denote the bipartite graph for the air marshal scheduling problem, $|A| = n_1, |B| = n_2$. Recall that G is a union of multiple isolated components, each containing all flights between two cities (see Figure 9.4). Within each component, we sort the flights in A by their *arrival* time and flights in B by their *departure* time.

We now show that generalized counting over the set of defender pure strategies admits a polynomial time algorithm. Our algorithm crucially exploits the following "order" structure.

Definition 6. *[Ordered Matching] In a bipartite graph $G = (A \cup B, E)$, a matching $M = \{e_1, \ldots, e_k\}$ is called an* ordered matching *if for any edge $e = (A_i, B_j)$ and $e' = (A_{i'}, B_{j'})$ in M, either $i > i'$, $j > j'$ or $i < i'$, $j < j'$.*

Visually, any two edges e, e' in an ordered matching satisfy that e is either "above" or "below" e' — they do not cross.

Since each flight has at most one air marshal, any assignment of air marshals must correspond to a matching in G. However, a pure strategy $\mathbf{e}$ — i.e., a set of covered flights — can be accomplished by different matchings. For example, the set $\mathbf{e} = \{A_1, A_2, B_1, B_2\}$ in Figure 9.4 can be achieved by the matching $\{(A_1, B_1), (A_2, B_2)\}$ or the matching $\{(A_1, B_2), (A_2, B_1)\}$. However, only the matching $\{(A_1, B_1), (A_2, B_2)\}$ is ordered. The following lemma shows that pure strategies and *ordered* k-matchings are in one-to-one correspondence.

Lemma 22. *In the air marshal scheduling problem, there exits an ordering of flights in A and B so that pure strategies and size-k ordered matchings are in one-to-one correspondence.*

Proof. It is easy to see that any ordered k-matching corresponds to one pure strategy. We prove the converse. Given any pure strategy S consisting of $2k$ flights, let $\widetilde{E} = \{e_1, \ldots, e_k\}$ be any matching that results in S. We claim that if there exist two edges $e, e' \in \widetilde{E}$ with $e = (A_i, B_j)$ and $e' = (A_{i'}, B_{j'})$ such that $i > i'$ and $j < j'$, then $(A_i, B_{j'})$ and $(A_{i'}, B_j)$ must also be edges in E. Since $e, e' \in \widetilde{E}$, we must have $T_1 < dep(B_j) - arr(A_i) < T_2$ and $T_1 < dep(B_{j'}) - arr(A_{i'}) < T_2$. Since flights in A are ordered increasingly by arrival time and flights in B are ordered increasingly by departure time, we have $arr(A_i) \geq arr(A_{i'})$ and $dep(B_j) \leq dep(B_{j'})$. These inequalities imply $dep(B_{j'}) - arr(A_i) \leq dep(B_{j'}) - arr(A_{i'}) \leq T_2$ and $dep(B_{j'}) - arr(A_i) \geq dep(B_j) - arr(A_i) \geq T_1$; therefore $(A_i, B_{j'}) \in E$. Similarly, one can show that $(A_j, B_{i'}) \in E$.

As a result, we can adjust the matching by using the edges $(A_i, B_{j'})$ and $(A_{i'}, B_j)$ instead. Such adjustments can continue until the matching becomes ordered. The procedure will terminate within a finite time by a simple potential function argument, with potential function $f(\widetilde{E}) = \sum_{e=(A_i,B_j)\in\widetilde{E}} |i-j|^2$. The above adjustment always strictly decreases the potential function since $|i-j|^2 + |i'-j'|^2 > |i-j'|^2 + |i'-j|^2$ if $i > i'$ and $j < j'$. The adjustment will terminate with an ordered matching and the ordered matching is unique, concluding our proof. □

Lemma 22 provides a way to reduce generalized counting over the set of pure strategies to generalized counting of size-k ordered matchings. Given any set of non-negative weights $\alpha \in \mathbb{R}_+^{n_1+n_2}$, we define edge weight $w_e = \alpha_{A_i}\alpha_{B_j}$ for any $e = (A_i, B_j) \in E$. As a result, the weight of any pure strategy equals the weight of the corresponding size-k ordered matching with edge weights w_e's.

Next we show that generalized counting of size-k ordered matchings admits an efficient algorithm. The main idea is to dynamically compute the generalized sum of size-k ordered matchings according to some "order" of the bipartite graphs. More specifically, define $E_{l,r} \subseteq E$ to be the set of edges that are "under" A_l and B_r, where $A_l \in A, B_r \in B$. Formally, any $e = (A_i, B_j) \in E$ is in $E_{l,r}$ iff $i \leq l, j \leq r$. We build a dynamic programing table with terms $\mathrm{DP}(l, r; d) = \sum_{M:\, M \subseteq E_{l,r}, |M|=d} w_M$, in which $\mathrm{DP}(l, r; d)$ is the sum of the weights of all size-d ordered matchings with edges in $E_{l,r}$. Now, to compute $\mathrm{DP}(l, r; d)$, we only need to enumerate all the possibilities of the uppermost edge in the ordered matching, given that $\mathrm{DP}(i, j; d)$s are known for $i < l$ and $j < r$. This can be done by a dynamic program (Algorithm 8). The correctness of Algorithm 8 follows by definition.

Algorithm 8: Generalized Counting of Ordered k-Matchings

Input: : $G = (A \cup B, E)$; $w_e \geq 0$ for any $e \in E$.

Output: : $\sum_{M:\, |M|=d} w_M$ $\quad$ % M is an ordered matching

1: **Initialization**: $\mathrm{DP}(l, r; 0) = 1$ for $l = 0, .., n_1, r = 0, .., n_2$; $\mathrm{DP}(0, r; d) = \mathrm{DP}(l, 0; d) = 0$ for all $d \geq 1, l = 0, 1, , \ldots, n_1, r = 0, 1, \ldots, n_2$.

2: Update: for $d = 1, \ldots, k, l = 2, \ldots, n_1,\ r = 2, \ldots, n_2$:

$$\mathrm{DP}(l, r; d) = T(l-1, r-1; d) + \sum_{\substack{e=(A_i, B_j) \in E_{l,r} \\ \text{s.t. } i=l \text{ or } j=r}} w_e \cdot \mathrm{DP}(i-1, j-1; d-1).$$

3: **return** $\mathrm{DP}(n_1, n_2; k)$.

Step 2

Let $\vec{\beta}^*$ be the optimal solution of CP (9.2) for the air marshal scheduling problem. Invoking Algorithm 8, we can compute $\vec{\beta}^*$ in $\mathrm{poly}(n, k)$ time. Let $\alpha_i = e^{-\beta_i^*}$ for all $i \in A \cup B$. Then the following algorithm (Algorithm 9) efficiently samples a pure strategy $\mathbf{e}$ from an exponentially large support with probability θ_e^* defined by Equation (9.3). The correctness of Algorithm 9 follows a similar argument as the proof of Lemma 21; we thus will not repeat the details here.

Algorithm 9: Max-entropy sampling in the air marshal scheduling problem

Input: : $\vec{\alpha} \in [0, \infty)^{n_1+n_2}$, k.

Output: : a pure strategy $\mathbf{e}$ using k air marshals.

Initialize: $\mathbf{e} = \emptyset$; build the DP table $T(l, r; d)$ as in the previous part.
Set $c = k$, $l = n_1, r = n_2$;
while $c > 0$ **do**
    Sampling: for any edge $e = (i, j) \in E_{l,r}$ incident on l or r, add edge e to $\mathbf{e}$ with probability
$$p = \frac{\alpha_i \alpha_j T(i-1, j-1; k-1)}{T(l, r; k)};$$
    if $e = (i, j)$ was added to $\mathbf{e}$ **then**
        $c = c - 1$;
        $l = i - 1, \;\; r = j - 1$
    else
        $l = l - 1, \;\; r = r - 1$
return $\rceil$.

9.3.2 Scalability Challenges and A Heuristic Sampling Algorithm

Though the sampling algorithm in Section 9.3.1 provably runs in polynomial time, the order of the polynomial is large due to repeated calls to the counting oracle. In fact, the algorithm can only scale to a problem size of about 300 flights. However, FAMS needs to schedule about $30,000$ flights every day. Therefore, it is necessary to develop a much more efficient algorithm in order to scale up to this real-world problem size.

In this section, we propose a heuristic sampling algorithm that matches any given marginal vector $\mathbf{x} \in \mathbf{conv}(\mathcal{E})$ and is expected to achieve high entropy. This algorithm works for *general* security games (not only the air marshal scheduling problem), and is *computationally efficient* as long as the underlying security game can be solved efficiently.

At a high level, our idea is to design a *randomized* implementation for the celebrated Carathéodory's theorem, which makes the following *existence* statement: for any bounded polytope $\mathcal{P} \subseteq \mathbb{R}^n$ and any $\mathbf{x} \in \mathcal{P}$, there exist (at most) $n + 1$ vertices of $\mathcal{P}$ such that $\mathbf{x}$ can be written as a convex combination of these vertices. Interpreting $\mathcal{P}$ as the convex hull of defender pure strategies, this means that any defender mixed strategy, i.e., a point in $\mathcal{P}$, can be decomposed as a distribution over at most $n + 1$ pure strategies (n is the number of targets). We turn this existence statement into an efficient randomized algorithm, named **CA**rathéodory **R**andomized **D**ecomposition (`CARD`).

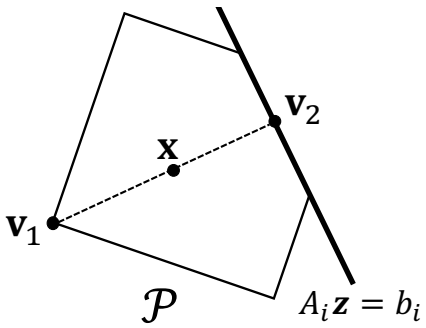


Figure 9.5: CARD Decomposition.

Consider any polytope $\mathcal{P} = \{\mathbf{z} : A\mathbf{z} \leq \mathbf{b}; M\mathbf{z} = \mathbf{c}\}$ explicitly represented by polynomially many linear constraints, and any $\mathbf{x} \in \mathcal{P}$. We use A_i, b_i to denote the i'th row of A and b respectively; $A_i\mathbf{z} = b_i$ is a *facet* of $\mathcal{P}$. Geometrically, CARD randomly picks a vertex $\mathbf{v}_1 = \arg\max_{\mathbf{z}\in\mathcal{P}}\langle \mathbf{a}, \mathbf{z}\rangle$ for a linear objective $\mathbf{a} \in [0,1]^n$ chosen *uniformly at random*. CARD then "walks along" the ray that originates from $\mathbf{v}_1$ and points to $\mathbf{x}$, until it crosses a facet of $\mathcal{P}$, denoted by $A_i\mathbf{z} = b_i$, at some point $\mathbf{v}_2$ (see the illustration in Figure 9.5). Thus, $\mathbf{x}$ can be decomposed as a convex combination of $\mathbf{v}_1, \mathbf{v}_2$. CARD then treats $\mathbf{v}_2$ as a new $\mathbf{x}$ and decomposes it within the facet $A_i\mathbf{z} = b_i$ recursively until $\mathbf{v}_2$ becomes a vertex. Details are presented in Algorithm 10.

Algorithm 10: CARD

Require: $\mathcal{P} = \{\mathbf{z} \subseteq \mathbb{R}^n : A\mathbf{z} \leq \mathbf{b}; M\mathbf{z} = \mathbf{c}\}$ and $\mathbf{x} \in \mathcal{P}$
Ensure: $\mathbf{v}_1, \ldots, \mathbf{v}_k$ and $p_1, \ldots, p_k$ such that $\sum_{i=1}^k p_i \cdot \mathbf{v}_i = \mathbf{x}$.

if rank(M) = n **then**
  Return the unique point $\mathbf{v}_1$ in $\mathcal{P}$ and $p_1 = 1$.
else
  Choose $\mathbf{a} \in [-1,1]^n$ *uniformly at random*.
  Compute $\mathbf{v}_1 = \arg\max_{\mathbf{z}\in\mathcal{P}}\langle \mathbf{a}, \mathbf{z}\rangle$.
  Compute $t = \min_{i:A_i(\mathbf{x}-\mathbf{v}_1)>0} \frac{b_i - A_i\mathbf{x}_i}{A_i(\mathbf{x}-\mathbf{v}_1)}$.
  Let i^* be the row achieving t, and $\mathcal{P}' = \{\mathbf{z} \in \mathcal{P} : A_{i^*}\mathbf{z} = b_{i^*}\}$.
  $\mathbf{v}_2 = \mathbf{x} + t(\mathbf{x} - \mathbf{v}_1)$; $p_1 = \frac{t}{t+1}$, $p_2 = \frac{1}{1+t}$.
  $[V', \mathbf{p}'] = \text{CARD}(\mathbf{v}_2, \mathcal{P}')$.
  return $V = (\mathbf{v}_1, V')$ and $\mathbf{p} = (p_1, p_2 \times \mathbf{p}')$.

A crucial ingredient of CARD is that each vertex $\mathbf{v}_1$ is the optimal vertex solution to a *uniformly random* linear objective. Recall that the max-entropy distribution over any given support under no constraints is the uniform distribution. The intuition underlying CARD is that these randomly selected vertices will inherit the high entropy of their linear objectives. Notice that the

decomposition generated by CARD is different in each execution due to its randomness; therefore the strategies generated by CARD are sampled from a very large support.

9.3.3 Experiments

We now experimentally compare MaxEn and CARD with traditional security game algorithms in the air marshal scheduling problem. We are not aware of any previous algorithm that directly computes the optimal defender strategy against a particular leakage model; therefore, the rigorously optimal solution is not available. We instead use a "harder" BaseLine which is the attacker utility assuming no leakage. This is the best (i.e., smallest) possible attacker utility. The most widely used approach for solving large-scale security games is the *column generation* technique (a.k.a., strategy/constraint generation (Jain et al., 2010; Bosansky, Jiang, Tambe, & Kiekintveld, 2015)). We compare MaxEn and CARD with ColG (the optimal mixed strategy computed via column generation[4] assuming no leakage). Note that without leakage, all three algorithms achieve the *same* solution quality since they implement the same marginal vector. The goal of this experiment is to test their robustness in the presence of information leakage.

Since it is impossible to obtain real-world data in this setting, all algorithms are thus tested on simulated instances for the Federal Air Marshal Scheduling problem with round-trip flights (FAMS). Like in the previous section, here we also generate zero-sum security games with reward and cost drawn randomly from $[0, 10]$ and $[-10, 0]$, respectively. All results are averaged over 20 games. In the tested instances, we assume that the attacker can monitor two randomly chosen outbound flights and seeks to attack one return flight.

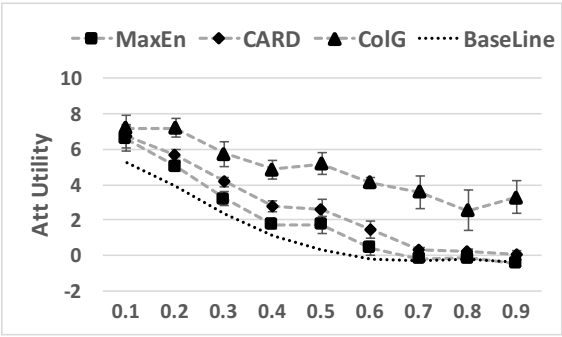


Figure 9.6: Utility comparisons in the FAMS domain (x-axis is the DtS ratio)

[4]The column generation technique is widely used in many security game algorithms. Though some security games use a compact linear program to directly compute the optimal marginal vector, the ultimate generation of a deployable mixed strategy still requires strategy generation techniques. In FAMS domain, ColG is precisely the ASPEN algorithm (Jain et al., 2010) — the leading algorithm today for scheduling air marshals at scale.

Figure 9.6 compares the defender utility obtained by different algorithms when there is information leakage. We vary the comparison on different deployment-to-saturation (DtS) ratios (Jain, Leyton-Brown, & Tambe, 2012). The DtS ratio captures the *fraction* of targets that can be covered in a pure strategy, which is $2k/n$ in the FAMS domain ($n = 60$ in Figure 9.6).

From Figure 9.6, we observe that `MaxEn` and `CARD` significantly outperforms `ColG` in our simulations; `CARD` is usually slightly outperformed by `MaxEn`. In fact, the attacker utility of the max-entropy approach is even close to the Baseline benchmark. This shows that the approach performs really well since the Baseline is the lowest possible attacker utility. We observed that the higher the DtS ratio is, the worse `ColG` performs. This is possibly because, with higher DtS, `ColG` quickly converges to an optimal mixed strategy with very small support, since each pure strategy covers many targets. Unfortunately, such a small-support strategy suffers severely from the curse of correlation. We observed that in FAMS games with 100 targets, `MaxEn`, `CARD`, `ColG` use 99997, 2199, 53 pure strategies on average, respectively (`MaxEn` samples 100,000 pure strategies in our experiments, and almost all of them are different).

9.4 The Design of Randomized Patrol Routes

In this section, we consider the problem of designing randomized patrol routes, as illustrated in Section 7.2. This setting belongs to a broader class of games termed *spatio-temporal* security games. These games are played out in space and time, and have applications in many domains, e.g., wildlife protection, protection of mobile ferries, etc. (Basilico, Gatti, & Amigoni, 2009a; Fang, Jiang, & Tambe, 2013; Yin, Xu, Gain, An, & Jiang, 2015). In these domains, the defender needs to move patrollers as time goes on. Due to the inherent correlation among the patroller's consecutive moves, these games are more likely to suffer from information leakage.

We start with the formal definition of the problem. Like most previous work, we focus on discretized spatio-temporal security games. Such games are described by a $T \times N$ grid graph

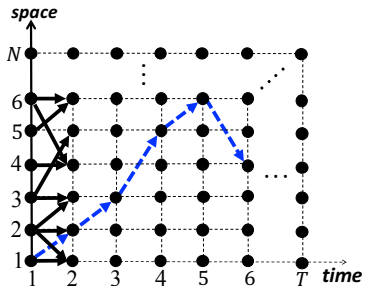


Figure 9.7: Structure of a spatio-temporal security game

$G = (V, E)$ indicating a problem with N cells and T time layers (see Figure 9.7). We use $v_{t,i}$ to denote a grid node, representing cell i at time t. Each $v_{t,i}$ is treated as a target, so there are $n = T \times N$ targets. The directed edges denote the patroller's feasible moves between cells within between consecutive time layers. Such feasibility usually incorporates speed limit, terrain constraints, etc. Figure 9.7 depicts some feasible moves between time layers 1 and 2. A feasible patrol path is a path in G starting from time 1 and ending at time T (e.g., the dashed path in Figure 9.7). Note that there are exponentially many patrol paths. We assume that the defender has k *homogeneous* patrollers, so a defender pure strategy corresponds to the *set of nodes* covered by k feasible patrol paths.

Different from the cases in the previous two sections, we will prove that it is computationally intractable in this setting to compute the distribution that maximizes entropy subject to matching a given marginal vector. We will then develop a polynomial-time algorithm for a well-motivated special case. Finally, we thoroughly evaluate the algorithm based on both synthetic and real-world data.

9.4.1 Complexity Barriers

We prove the following theorem in this section.

Theorem 9.4.1. *It is #P-hard to sample the max-entropy distribution for spatio-temporal security games even when there are only* two *time layers (i.e., $T = 2$).*

Proof. When there are two time steps, the game structure corresponds to a bipartite graph ($T = 2$ in Figure 9.7). It is important to notice that a pure strategy here does *not* simply correspond to a bipartite matching of size k; therefore we cannot reduce from the problem of counting size-k matchings. This is because the selected k edges are allowed to share nodes. Moreover, our definition of a pure strategy is the set of covered targets, not the edges themselves. In fact, sometimes one pure strategy can be achieved by different sets of k edges.

To prove the theorem, we reduce from the problem of counting bases of a transversal matroid, which is known to be #P-complete (Colbourn, Provan, & Vertigan, 1995). Given any bipartite graph $G = (L \cup R, E)$ with $|L| = k$, $|R| = n$ and $k \leq n$, any set $T \subseteq R$ is an independent set of the transversal matroid $\mathcal{M}(G)$ of G if there exists a matching of size $|T|$ in the subgraph induced by $L \cup T$; such a T is a base if $|T| = |L| = k$.

Given any bipartite graph $G = (L \cup R, E)$ with $k = |L| \leq |R| = n$, we reduce counting bases of the transversal matroid $\mathcal{M}(G)$ to computing the max-entropy distribution for the two-time-layer spatio-temporal security game on graph G.[5] Let $\mathcal{S}_{2k}$ denote the set of pure strategies

[5]Though the definition of spatio-temporal security games requires that each time layer has the same number of nodes, this requirement is not essential since one can always add *isolated* nodes to each time layer to equalize the number of nodes.

that cover exactly $2k$ nodes. We first reduce counting bases of $\mathcal{M}(G)$ to counting $\mathcal{S}_{2k}$. This is simply because if a pure strategy covers $2k$ nodes, it must cover all k nodes in L and another k nodes in R, and these $2k$ nodes are matchable.Then elements in $\mathcal{S}_{2k}$ and $\mathcal{M}(G)$ are in one-to-one correspondence.

Since counting reduces to generalized counting, we finally reduce generalized counting over the set $\mathcal{S}_{2k}$ to computing the max-entropy distribution for the following special subset of marginal vectors $\mathcal{X}_{2k} = \{\mathbf{x} \in [0,1]^{k+n} : x_{1,i} = 1, \forall i \in L; \sum_{i=1}^{n} x_{2,i} = k\}$. It is easy to see that any mixed strategy that matches a marginal vector $\mathbf{x} \in \mathcal{X}_{2k}$ must have support in $\mathcal{S}_{2k}$. Therefore, when considering the max-entropy distribution for any $\mathbf{x} \in \mathcal{X}_{2k}$, we can w.l.o.g. restrict the set of pure strategies to be $\mathcal{S}_{2k}$. By the computational equivalence between generalized counting and max-entropy sampling (Singh & Vishnoi, 2013), generalized counting over $\mathcal{S}_{2k}$ reduces to computing the max-entropy distribution for any $\mathbf{x} \in \mathcal{X}_{2k}$.

□

9.4.2 An Efficient Algorithm for a Restricted Setting

Theorem 9.4.1 suggests that it is unlikely that there is an efficient algorithm for sampling the max-entropy distribution with given marginal probabilities in this setting. Moreover, there is no known *polynomial size* compact formulation for sampling the max-entropy distribution. Thus we cannot utilize state-of-the-art optimization software to tackle the problem neither.

Nevertheless, in this section, we show that the max-entropy approach can be efficiently implemented in a well-motivated special setting where the defender only possesses a small number of patrollers. For example, in wildlife protection, the defender usually has only one or two patrol teams at each patrol post (Fang et al., 2016a); the US Coast Guard uses two patrollers to protect Staten Island ferries (Fang et al., 2013). We show that when the number of patrollers is small (i.e., a constant), the max entropy distribution can be sampled efficiently.

Theorem 9.4.2. *When the number of patrollers is a constant, there is* $\mathrm{poly}(N, T)$ *time algorithm for sampling the distribution that maximizes entropy subject to matching any given marginal vector in spatio-temporal security games.*

Similar to the proof of Theorem 9.3.1, this proof also has two steps:

- First, we design a $\mathrm{poly}(N, T)$ time algorithm for the generalized counting problem over the set system $\mathcal{E}$ of defender pure strategies.
- Second, we will design an efficient sampling algorithm that samples a pure strategy $\mathbf{e}$ from an exponentially large support with probability θ_e^* defined by Equation (9.3).

Step 1

For ease of presentation, our description focuses on the case with two patrollers, though it easily generalizes to a constant number of patrollers. We propose a dynamic program (DP) that exploits the natural chronological order of the targets along the temporal dimension.[6]

Let us call a pure strategy a 2-*path*, since each of the two patrollers takes a path on G. Our goal is to compute the weighted count of all 2-paths in a grid graph G, each weighted by the product of the node weights it traverses. Our goal is to compute the weighted count of combinations of two paths in G (one for each patroller), where the weight is the product of the node weights that the two paths traverse. Let $\{\alpha_{t,i}\}_{t\in[T],i\in[n]}$ be any given weight set. Obviously, the counting problem is easy if $T = 1$, i.e., only one time layer. Our key observation is that the solution for $T = t$ can be constructed by utilizing the solutions for $T = t - 1$. For any $1 \le i \le j \le N$, we use $\mathrm{DP}(i, j; t)$ to denote the solution to the counting problem restricted to the truncated graph with only time layers $1, 2, \cdots, t$, satisfying that the two patroller must end at cells i, j at time t. Observe that $\mathrm{DP}(i, j; 1) = \alpha_{1,i}\alpha_{1,j}$ when $i \neq j$ and $\mathrm{DP}(i, i; 1) = \alpha_{1,i}$. We then use the following update rule for $t \ge 2$:

$$\mathrm{DP}(i,j;t) = \begin{cases} \alpha_{t,i}\alpha_{t,j} \cdot \sum_{(i',j')\in \mathrm{pre}(i,j)} \mathrm{DP}(i',j';t-1) & \text{if } i < j \\ \alpha_{t,i} \cdot \sum_{(i',j')\in \mathrm{pre}(i,j)} \mathrm{DP}(i',j';t-1) & \text{if } i = j \end{cases} \tag{9.4}$$

where $\mathrm{pre} = \{(i', j') : i' \le j' \, s.t. \, v_{t-1,i'}, v_{t-1,j'} \text{ can reach } v_{t,i}, v_{t,j} \}$ is essentially the set of all pairs of nodes that can reach $v_{t,i}, v_{t,j}$. Note that the solution to the generalized counting problem is $\sum_{i\le j} \mathrm{DP}(i, j; T)$. The correctness of the algorithm follows from the observation that if the two patrollers are at $v_{t,i}$ and $v_{t,j}$, they must come from $v_{t-1,i'}$ and $v_{t-1,j'}$ for certain $(i', j') \in \mathrm{pre}(i, j)$. The updating rule simply aggregates all such choices. The algorithm runs in $\mathrm{poly}(N, T)$ time.

Step 2

Let $\vec{\beta}^*$ be the optimal solution of CP (9.2) in the spatio-temporal setting. Let $\alpha_{t,i} = e^{-\beta^*_{t,i}}$ for all t, i. Then the following algorithm efficiently samples a pure strategy $\mathbf{e}$ from an exponentially large support with probability θ^*_e defined by Equation (9.3). The correctness of Algorithm 11 follows from a similar argument as the proof of Lemma 21.

[6] Dynamic programming is widely used in counting problems. See, e.g., (Cryan & Dyer, 2002; Dyer, 2003) as well as the remarks in (Valiant, 1979). The novel parts usually lie at careful analysis of the problem to uncover the proper structure for DP.

Algorithm 11: Max-Entropy Sampling In Spatio-Temporal Security Games

Input: : $\vec{\alpha} \in [0, \infty)^{(T+1)\times(N+1)}$.

Output: : a pure strategy $\mathbf{e}$.

1: Initialize: $\mathbf{e} = \emptyset$; build the DP table $DP(i, j; t)$ according to Equation (9.4).

2: Sample two nodes $(v_{i,T}, v_{j,T})$, with $0 \le i < j \le N$, at time T with probability

$$p = \frac{DP(i, j; T)}{\sum_{i=0}^{N} \sum_{j=i}^{N} DP(i, j; T)};$$

Let i^*, j^* be the two sampled nodes; Add them to $\mathbf{e}$.

3: Define $a = i^*, b = j^*$.

4: **for** $t = T - 1$ to 0 **do**

5: Sample nodes $v_{t,i}, v_{t,j}$, for $(i, j) \in pred(a, b)$ and $0 \le i \le j \le N$, with probability

$$p = \frac{\alpha_{t,i}\alpha_{t,j} DP(i, j; t)}{DP(a, b; t + 1)} \qquad \left(p = \frac{\alpha_{t,i} DP(i, j; t)}{DP(a, b; t + 1)} \text{ if } i = j\right);$$

6: Let v_{t,i^*}, v_{t,j^*} be the sampled nodes above, and add v_{t,i^*}, v_{t,j^*} to $\mathbf{e}$

7: Update $a = i^*, b = j^*$.

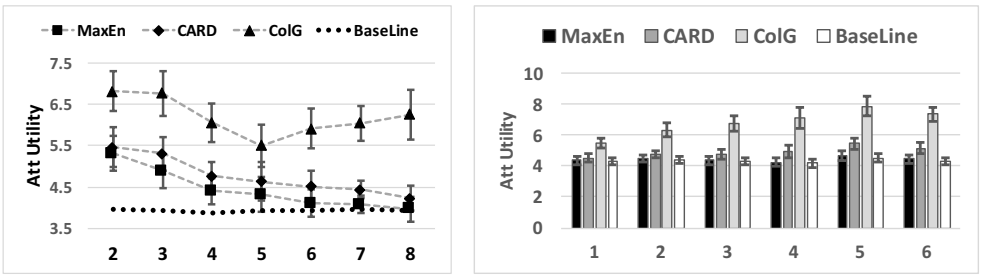
8: **return** $\mathbf{e}$.

9.4.3 Experiments

9.4.3.1 Synthetic Data

We first experimentally compare `MaxEn`, `CARD` with traditional algorithms for spatio-temporal security games. Like the setup in Section 9.3.3, we are not aware of any previous algorithm that directly computes the optimal defender strategy against a particular leakage model. Instead we use a "harder" `BaseLine` which is the attacker utility assuming no leakage. This is the best (i.e., smallest) possible attacker utility. We compare `MaxEn` and `CARD` with `ColG` (the optimal mixed strategy computed via column generation assuming no leakage). Note that without leakage, all three algorithms achieve the *same* solution quality since they implement the same marginal vector. The goal of this experiment is to test their robustness in the presence of information leakage.

In this part, we will test all algorithms on simulated instances. All results are averaged over 20 zero-sum security games with utilities drawn randomly from $[-10, 10]$. In the tested instances, unless specifically mentioned, we always assume that the attacker can monitor two randomly chosen targets at the first time layer (i.e., $t = 1$) and seeks to attack one target at the last time layer (i.e., $t = T$).



(a) Increase T

(b) Increase #MoT

Figure 9.8: Utility comparisons in spatio-temporal security games.

Figure 9.8 compares the defender utility obtained by different algorithms when there is information leakage. From the figure, it is easy to see that `MaxEn` and `CARD` significantly outperform `ColG`; `CARD` is usually slightly outperformed by `MaxEn`. Moreover, the attacker utility that the max-entropy approach induces is even close to the Baseline benchmark. This shows that the approach performs very well in the simulated random games since the Baseline is the lowest possible attacker utility.

Figure 9.8(a) compares the algorithms by varying the number of time layers T, but fixing $N = 9$. When T increases, `MaxEn` and `CARD` approach the `BaseLine`, i.e., the lowest possible attacker utility. This shows that in patrolling strategies of large entropy, the correlation between a patroller's initial and later moves gradually disappears as time goes on. This illustrates the validity of the max-entropy approach for mitigating CoC. In Figure 9.8(b), we fix $T = 9, N = 9$, and compare the algorithms by varying the number of monitored targets (#MoT). We observed that even when the attacker can monitor 6 out of 9 targets at $t = 1$, `MaxEn` and `CARD` are still close to `BaseLine`, while the performance of `ColG` gradually decrease as #MoT increases.

9.4.3.2 Real-World Data from the Queen Elizabeth National Park

Finally, we test our algorithm on a real-world wildlife crime dataset from Uganda's Queen Elizabeth Protected Area (QEPA). QEPA spans approximately 2,520 square kilometers and is patrolled by wildlife park rangers. While on patrol, they collect data on animal sightings and signs of illegal human activity (e.g., poaching, trespassing). In addition to this observational data, the dataset contains terrain information (e.g., slope, vegetation), distance data (e.g., nearest patrol post), animal density, and the kilometers walked by rangers in an area (i.e., effort).

There are 39 patrol posts at QEPA. We test the patrol design algorithm on the real data/model at patrol posts 11, 19 and 24, which are the three posts that had the most attacks in the three months of our testing. We divide the area around each patrol post into 1 square kilometer grid cells and optimize the patrol route for that particular post based on the importance of each cell estimated from several features (e.g., animal density, past captures, terrain, past effort). In our

data, all posts have less than 100 cells/targets reachable from the post by a route of maximum duration $T = 12$ (equivalently, a 12-cells long route).

We aim at comparing our patrol design algorithm with the real patrol routes adopted in the past by park rangers. One major challenge in experimenting with this real data is the lack of ground truth. In particular, for the past patrolling, we do not know what happened at those cells that were not patrolled nor doe we know what would have happened if the rangers had adopted our algorithm. Therefore, as an approximation, we use the state-of-the-art predictive model in (Gholami, Ford, Fang, Plumptre, Tambe, Driciru, Wanyama, Rwetsiba, Nsubaga, & Mabonga, 2017) to estimate the attacks at each cell. This is of course not perfect, but it is the best comparison we could do currently since (Gholami et al., 2017) shows that this predictive model outperforms all previous poaching prediction models and provides relatively accurate predictions on the QEPA dataset.

The comparisons are conducted under the following criteria:

- **#Detection**: total number of detected attacks under the prediction model. Since the prediction model we adopt is a 0-1 classification algorithm, in this case **#Detection** also equals the number of cells at which the corresponding patrol routes result in detected attacks.
- **#Routes**: the number of different patrol routes in 90-day route samples (corresponding to a 3-month patrolling period).
- **Entropy**: The entropy of the empirical distribution of the 90 samples.

The first criterion concerns the efficacy of the patrol routes while the last two criteria are used to test the unpredictability of the patrol routes. For the **#Detection** criterion, a/b means that out of the b cells with predicted attacks, a are patrolled. For example, in Table 9.1, the "15/19" means the following: 19 cells are predicted to be attacked; the patrol route visits 15 of these 19 cells. A higher value of **#Routes** means that the patroller has more choices of patrol routes, and thus less explorable by the poacher; **Entropy** is a natural measure to quantify uncertainty.

Criteria	Post 11			Post 19			Post 24	
	`MaxEn`	Past		`MaxEn`	Past		`MaxEn`	Past
#Detections	15/19	4/19		6/6	5/6		4/4	3/4
#Routes	61	4		22	33		34	5
Entropy	4.0	1.2		2.6	3		2.8	1.4

Table 9.1: Comparisons of different criteria at different patrol posts

The results are jointly presented in Table 9.1. As we can see, the patrol routes generated by `MaxEn` clearly outperform past patrolling in terms of the **#Detections** criteria. The routes

we generate can detect most (if not all) of the predicted attacks. In terms of unpredictability, past patrolling does not have stable performance. Particularly, it follows only a few routes at posts 11 and 24 with low unpredictability but takes many different routes at post 19 with high unpredictability. This is a consequence of various factors at different posts, e.g., the patroller's preferences, location of the patrol post (e.g., inside or at the boundary of the area), terrain features, etc. On the other hand, `MaxEn` always comes with sufficient unpredictability. This shows the advantage of `MaxEn` over the past patrolling.

Part IV

Conclusion

Chapter 10

Conclusions and Open Directions

This thesis seeks to understand how information affects agents' decision making in strategic interactions through a computational lens. It illustrates the double-edged role of information through two threads of research: (1) how to utilize information to one's own advantage in strategic interactions; (2) how to mitigate losses resulting from information leakage to an adversary. We conduct both theoretical study to understand the algorithmic foundations of these problems as well as applied study to show how these problems can be modeled and solved in real-world applications. Notably, algorithms from this thesis have been implemented and tested in the field by security agencies. This shows the potential real-world impact of understanding the role of information in decision making. Though the work of this thesis is primarily motivated by the strategic interaction between security agencies and adversaries (i.e., security games), we believe that the foundational economic models we studied and the basic tools we developed can find applications in many other application domains as well.

We conclude with some future directions that are motivated by this thesis or are aligned with its theme.

> **Future Direction I:** *How to optimally persuade receivers in more realistic, yet more intricate, settings by taking into account, e.g., externalities among receivers, uncertainties in receivers' beliefs, and multi-round interactions between the sender and receivers? Can we still design efficient algorithms for these problems?*

In Section 5.2, we provide a thorough algorithmic analysis for two of the most foundational models of persuasion, and consider the setting where there is either one receiver or multiple receivers with binary actions and no externalities. However, in many settings, the receivers may have externalities and their decisions affect each others' payoffs. Examples include: (1) auction settings where the auctioneer (the sender) may want to persuade bidders (receivers) about the value of the item for sale; (2) traffic routing where certain recommendation systems like Google Maps (the sender) may want to persuade drivers (receivers) about choices of routing paths; (3)

voting settings where a principal (the sender) may want to persuade voters (receivers) regarding which candidate to vote for. In all these applications, the receivers' decisions will affect each other's payoffs. One fundamental problem is to understand the complexity of persuasion due to such externalities.

One important concern about previous models of persuasion is that the sender and receivers must share the same prior belief and the receivers must know precisely the signaling scheme. However, in practice, these beliefs usually come from observations or data analysis, and thus are rarely precise. To make these models more realistic, we have to understand how robust the models or computation are to uncertain or imprecise player beliefs. Moreover, in many settings, persuasion is done in a multi-round interaction between the sender and receiver. How to optimally persuade receivers in a multi-round interaction and how would this change the complexity of the problem?

> **Future Direction II:** *What are other roles that information could play in security domains? How to improve security decision making by taking them into account?*

This thesis initiates a computational study of the role of information in security decision making and considers how to utilize the defender's informational advantage and how to deal with harms due to information leakage. This motivates the future study of many other possible roles that information could play in security domains. For example, in security games we usually assume that the adversary will surveil the defender's strategy and then strategically respond. However, in many domains, the defender can also surveil the adversary by using surveillance tools like closed-circuit televisions. One important question in these problems is how to integrate such information into the defender's decision making to improve the defense. This thesis studies how the defender can utilize informational advantages to deceive the adversary. However, in practice, the adversary can also be deceptive and may provide misleading information to the defender. Therefore, another interesting question is how to take such strategic adversary behavior into account and how it would affect the defender's decision making. This question is particularly relevant in cybersecurity domains since the attackers there are usually more sophisticated, and deceptive attacks have been frequently observed in practice (Rowe & Rothstein, 2004; Rowe, 2006). These are just two examples, and there are many other questions pertaining to the role of information in security domains that remain largely unexplored. More importantly, when considering a particular application, how can we take into account various effects of information together to make better decisions?

> **Future Direction III:** *Besides security domains, how does information affect the decision making in other multi-agent systems and applications? How to model and design efficient algorithms for these applications?*

This thesis is primarily motivated by strategic interactions in security domains. This, however, is just one particular example of multi-agent systems. In the future, it will be interesting to study other systems with self-interested agents, and our work indicates that information can play a crucial role in influencing the outcomes of these systems. This broad line of research is particularly relevant in the digital age since the ubiquitous access to data and advances in data analytics have made it much easier today to generate and communicate information. This is profoundly affecting people's decision making. Indeed, many of our decisions today (e.g., which route to take, which restaurant to go to, which stock to invest in, which candidate to vote for, etc.) are affected by, or even rely on, numerous information sources such as news, media, social networks, search engines and various recommendation system applications (e.g., Google Maps, Yelp, etc.). This brings tremendous opportunities for studying the effects of information in these domains and understanding how we can utilize these effects to improve decision making. Moreover, we believe that the requirement of automated applications today makes it particularly suitable to develop computational techniques and algorithms for solving these problems.

Part V

Appendices

Appendix A

Omitted Proofs From Section 5.2

A.1 Omissions from Section 5.2.2

A.1.1 Symmetry of the Optimal Scheme (Theorem 5.2.1)

To prove Theorem 5.2.1, we need two closure properties of optimal signaling schemes — with respect to permutations and convex combinations. We use π to denote a permutation of $[n]$, and let SS_n denote the set of all such permutations. We define the permutation $\pi(\theta)$ of a state of nature $\theta \in [m]^n$ so that $(\pi(\theta))_j = \theta_{\pi(j)}$, and similarly the permutation of a signal σ_i so that $\pi(\sigma_i) = \sigma_{\pi(i)}$. Given a signature $\mathcal{M} = \{(M^{\sigma_i}, \sigma_i)\}_{i\in[n]}$, we define the permuted signature $\pi(\mathcal{M}) = \{(\pi M^{\sigma_i}, \pi(\sigma_i))\}_{i\in[n]}$, where πM denotes applying permutation π to the rows of a matrix M.

Lemma 23. *Assume the action payoffs are i.i.d., and let $\pi \in SS_n$ be an arbitrary permutation. If $\mathcal{M}$ is the signature of a signaling scheme φ, then $\pi(\mathcal{M})$ is the signature of the scheme φ_π defined by $\varphi_\pi(\theta) = \pi(\varphi(\pi^{-1}(\theta)))$. Moreover, if φ is persuasive and optimal, then so is φ_π.*

Proof. Let $\mathcal{M} = \{(M^\sigma, \sigma)\}_{\sigma\in\Sigma}$ be the signature of φ, as given in the statement of the lemma. We first show that $\pi(\mathcal{M}) = \{(\pi M^\sigma, \pi(\sigma))\}_{\sigma\in\Sigma}$ is realizable as the signature of the scheme φ_π.

By definition, it suffices to show that $\sum_\theta \lambda(\theta)\varphi_\pi(\theta, \pi(\sigma))M^\theta = \pi M^\sigma$ for an arbitrary signal $\pi(\sigma)$.

$$\begin{aligned}
\sum_\theta \lambda(\theta)\varphi_\pi(\theta, \pi(\sigma))M^\theta &= \sum_\theta \lambda(\theta)\varphi(\pi^{-1}(\theta), \sigma)M^\theta && \text{(by definition of } \varphi_\pi\text{)} \\
&= \pi \sum_{\theta\in\Theta} \lambda(\theta)\varphi(\pi^{-1}(\theta), \sigma)(\pi^{-1}M^\theta) && \text{(by linearity of permutation)} \\
&= \pi \sum_{\theta\in\Theta} \lambda(\theta)\varphi(\pi^{-1}(\theta), \sigma)M^{\pi^{-1}(\theta)} \\
&= \pi \sum_{\theta\in\Theta} \lambda(\pi^{-1}(\theta))\varphi(\pi^{-1}(\theta), \sigma)M^{\pi^{-1}(\theta)} && \text{(Since } \lambda \text{ is i.i.d.)} \\
&= \pi \sum_{\theta'\in\Theta} \lambda(\theta')\varphi(\theta', \sigma)M^{\theta'} && \text{(by renaming } \pi^{-1}(\theta) \text{ to } \theta'\text{)} \\
&= \pi M^\sigma && \text{(by definition of } M^\sigma\text{)}
\end{aligned}$$

Now, assuming φ is persuasive, we check that φ_π is persuasive by verifying the relevant inequality for its signature.

$$\boldsymbol{\rho} \cdot (\pi M^{\sigma_i})_{\pi(i)} - \boldsymbol{\rho} \cdot (\pi M^{\sigma_i})_{\pi(j)} = \boldsymbol{\rho} \cdot M_i^{\sigma_i} - \boldsymbol{\rho} \cdot M_j^{\sigma_i} \geq 0$$

Moreover, we show that the sender's utility is the same for φ and φ_π, completing the proof.

$$\boldsymbol{\xi} \cdot (\pi M^{\sigma_i})_{\pi(i)} = \boldsymbol{\xi} \cdot (M^{\sigma_i})_i$$

□

Lemma 24. *Let $t \in [0, 1]$. If $\mathcal{A} = (A^{\sigma_1}, \ldots, A^{\sigma_n})$ is the signature of scheme φ_A, and $\mathcal{B} = (B^{\sigma_1}, \ldots, B^{\sigma_n})$ is the signature of a scheme φ_B, then their convex combination $\mathcal{C} = (C^{\sigma_1}, \ldots, C^{\sigma_n})$ with $C^{\sigma_i} = tA^{\sigma_i} + (1-t)B^{\sigma_i}$ is the signature of the scheme φ_C which, on input θ, outputs $\varphi_A(\theta)$ with probability t and $\varphi_B(\theta)$ with probability $1-t$. Moreover, if φ_A and φ_B are both optimal and persuasive, then so is φ_C.*

Proof. This follows almost immediately from the fact that the optimization problem in Figure 5.2 is a linear program, with a convex feasible set and a convex family of optimal solutions. We omit the straightforward details. □

Proof of Theorem 5.2.1

Given an optimal and persuasive signaling scheme φ with signature $\{(M^{\sigma_i}, \sigma_i)\}_{i\in[n]}$, we show the existence of a symmetric optimal and persuasive scheme of the form in Definition 1. According to Lemma 23, for $\pi \in SS_n$ the signature $\{(\pi M^{\sigma_i}, \pi(\sigma_i))\}_{i\in[n]}$ — equivalently written as

$\{(\pi M^{\sigma_{\pi^{-1}(i)}}, \sigma_i\}_{i\in[n]}$ — corresponds to the optimal persuasive scheme φ_π. Invoking Lemma 24, the signature

$$\{(A^{\sigma_i}, \sigma_i)\}_{i\in[n]} = \{(\frac{1}{n!}\sum_{\pi\in SS_n} \pi M^{\sigma_{\pi^{-1}(i)}}, \sigma_i)\}_{i\in[n]}$$

also corresponds to an optimal and persuasive scheme, namely the scheme which draws a permutation π uniformly at random, then signals according to φ_π.

Observe that the ith row of the matrix $\pi M^{\sigma_{\pi^{-1}(i)}}$is the $\pi^{-1}(i)$th row of the matrix $M^{\sigma_{\pi^{-1}(i)}}$. Expressing $A_i^{\sigma_i}$ as a sum over permutations $\pi \in SS_n$, and grouping the sum by $k = \pi^{-1}(i)$, we can write

$$\begin{aligned}
A_i^{\sigma_i} &= \frac{1}{n!}\sum_{\pi\in SS_n} [\pi M^{\sigma_{\pi^{-1}(i)}}]_i \\
&= \frac{1}{n!}\sum_{\pi\in SS_n} M_{\pi^{-1}(i)}^{\sigma_{\pi^{-1}(i)}} \\
&= \frac{1}{n!}\sum_{k=1}^{n} M_k^{\sigma_k} \cdot \left|\left\{\pi \in SS_n : \pi^{-1}(i) = k\right\}\right| \\
&= \frac{1}{n!}\sum_{k=1}^{n} M_k^{\sigma_k} \cdot (n-1)! \\
&= \frac{1}{n}\sum_{k=1}^{n} M_k^{\sigma_k},
\end{aligned}$$

which does not depend on i. Similarly, the jth row of the matrix $\pi M^{\sigma_{\pi^{-1}(i)}}$ is the $\pi^{-1}(j)$th row of the matrix $M^{\sigma_{\pi^{-1}(i)}}$. For $j \neq i$, expressing $A_j^{\sigma_i}$ as a sum over permutations $\pi \in SS_n$, and grouping the sum by $k = \pi^{-1}(i)$ and $l = \pi^{-1}(j)$, we can write

$$\begin{aligned}
A_j^{\sigma_i} &= \frac{1}{n!}\sum_{\pi\in SS_n} [\pi M^{\sigma_{\pi^{-1}(i)}}]_j \\
&= \frac{1}{n!}\sum_{\pi\in SS_n} M_{\pi^{-1}(j)}^{\sigma_{\pi^{-1}(i)}} \\
&= \frac{1}{n!}\sum_{k\neq l} M_l^{\sigma_k} \cdot \left|\left\{\pi \in SS_n : \pi^{-1}(i) = k, \pi^{-1}(j) = l\right\}\right| \\
&= \frac{1}{n!}\sum_{k\neq l} M_l^{\sigma_k} \cdot (n-2)! \\
&= \frac{1}{n(n-1)}\sum_{k\neq l} M_l^{\sigma_k},
\end{aligned}$$

which does not depend on i or j. Let

$$\begin{aligned} \boldsymbol{x} &= \frac{1}{n}\sum_{k=1}^{n} M_k^{\sigma_k}; \\ \boldsymbol{y} &= \frac{1}{n(n-1)}\sum_{k\neq l} M_l^{\sigma_k}. \end{aligned}$$

The signature $\{(A^{\sigma_i}, \sigma_i)\}_{i\in[n]}$ therefore describes an optimal, persuasive, and symmetric scheme with s-signature $(\boldsymbol{x}, \boldsymbol{y})$.

A.1.2 The Optimal Scheme

Proof of Lemma 1

For the "only if" direction, $||\boldsymbol{x}||_1 = \frac{1}{n}$ and $\boldsymbol{x} + (n-1)\boldsymbol{y} = \boldsymbol{q}$ were established in Section 5.2.2. To show that $\boldsymbol{\tau}$ is a realizable symmetric reduced form for an allocation rule, let φ be a signaling scheme with s-signature $(\boldsymbol{x}, \boldsymbol{y})$. Recall from the definition of an s-signature that, for each $i \in [n]$, signal σ_i has probability $1/n$, and $n\boldsymbol{x}$ is the posterior distribution of action i's type conditioned on signal σ_i. Now consider the following allocation rule: Given a type profile $\theta \in [m]^n$ of the n bidders, allocate the item to bidder i with probability $\varphi(\theta, \sigma_i)$ for any $i \in [n]$. By Bayes rule,

$$\begin{aligned} \mathbf{Pr}[i \text{ gets item} | i \text{ has type } j] &= \mathbf{Pr}[i \text{ has type } j | i \text{ gets item}] \cdot \frac{\mathbf{Pr}[i \text{ gets item}]}{\mathbf{Pr}[i \text{ has type } j]} \\ &= nx_j \cdot \frac{1/n}{q_j} = \frac{x_j}{q_j} \end{aligned}$$

Therefore $\boldsymbol{\tau}$ is indeed the reduced form of the described allocation rule.

For the "if" direction, let $\boldsymbol{\tau}$, $\boldsymbol{x}$, and $\boldsymbol{y}$ be as in the statement of the lemma, and consider an allocation rule A with symmetric reduced form $\boldsymbol{\tau}$. Observe that A always allocates the item, since for each player $i \in [n]$ we have $\mathbf{Pr}[i \text{ gets the item}] = \sum_{j=1}^{m} q_j\tau_j = \sum_{j=1}^{m} x_j = \frac{1}{n}$. We define the direct signaling scheme φ_A by $\varphi_A(\theta) = \sigma_{A(\theta)}$. Let $\mathcal{M} = (M^{\sigma_1}, \ldots, M^{\sigma_n})$ be the signature of φ_A. Recall that, for $\theta \sim \lambda$ and arbitrary $i \in [n]$ and $j \in [m]$, $M_{ij}^{\sigma_i}$ is the probability that $\varphi_A(\theta) = \sigma_i$ and $\theta_i = j$; by definition, this equals the probability that A allocates the item to player i and her type is j, which is $\tau_j q_j = x_j$. As a result, the signature $\mathcal{M}$ of φ_A satisfies $M_i^{\sigma_i} = \boldsymbol{x}$ for every action i. If φ_A were symmetric, we would conclude that its s-signature is $(\boldsymbol{x}, \boldsymbol{y})$ since every s-signature $(\boldsymbol{x}, \boldsymbol{y}')$ must satisfy $\boldsymbol{x} + (n-1)\boldsymbol{y}' = \boldsymbol{q}$ (see Section 5.2.2). However, this is not guaranteed when the allocation rule A exhibits some asymmetry. Nevertheless, φ_A can be "symmetrized" into a signaling scheme φ'_A which first draws a random permutation $\pi \in SS_n$, and signals $\pi(\varphi_A(\pi^{-1}(\theta)))$. That φ'_A has s-signature $(\boldsymbol{x}, \boldsymbol{y})$ follows a similar argument to that used in the proof of Theorem 5.2.1, and we therefore omit the details here.

Finally, observe that the description of φ'_A above is constructive assuming black-box access to A, with runtime overhead that is polynomial in n and m.

Proof of Lemma 2

By Lemma 1, we can re-write LP (5.2) as follows:

$$\begin{aligned} \text{maximize} \quad & n\boldsymbol{\xi} \cdot \boldsymbol{x} \\ \text{subject to} \quad & \boldsymbol{\rho} \cdot \boldsymbol{x} \geq \boldsymbol{\rho} \cdot \boldsymbol{y} \\ & \boldsymbol{x} + (n-1)\boldsymbol{y} = \boldsymbol{q} \\ & ||\boldsymbol{x}||_1 = \tfrac{1}{n} \\ & (\tfrac{x_1}{q_1}, ..., \tfrac{x_m}{q_m}) \text{ is a realizable symmetric reduced form} \end{aligned} \tag{A.1}$$

From (Border, 1991, 2007; Cai et al., 2012; Alaei et al., 2012), we know that the family of all the realizable symmetric reduced forms constitutes a polytope, and moreover that this polytope admits an efficient separation oracle. The runtime of this oracle is polynomial in m and n, and as a result the above linear program can be solved in $poly(n, m)$ time using the Ellipsoid method.

A.1.3 A Simple $(1 - 1/e)$-Approximate Scheme

Proof of Theorem 5.2.3

Given a binary signal $\sigma = (o_1, \ldots, o_n) \in \{\mathbf{HIGH}, \mathbf{LOW}\}^n$, the posterior type distribution for an action equals $n\boldsymbol{x}^*$ if the corresponding component signal is $\mathbf{HIGH}$, and equals $n\boldsymbol{y}^*$ if the component signal is $\mathbf{LOW}$. This is simply a consequence of the independence of the action types, the fact that the different component signals are chosen independently, and Bayes' rule. The constraint $\boldsymbol{\rho} \cdot \boldsymbol{x}^* \geq \boldsymbol{\rho} \cdot \boldsymbol{y}^*$ implies that the receiver prefers actions i for which $o_i = \mathbf{HIGH}$, any one of which induces an expected utility of $n\boldsymbol{\rho} \cdot \boldsymbol{x}^*$ for the receiver and $n\boldsymbol{\xi} \cdot \boldsymbol{x}^*$ for the sender. The latter quantity matches the optimal value of LP (5.3). The constraint $||\boldsymbol{x}||_1 = \frac{1}{n}$ implies that each component signal is $\mathbf{HIGH}$ with probability $\frac{1}{n}$, independently. Therefore, the probability that at least one component signal is $\mathbf{HIGH}$ equals $1 - (1 - \frac{1}{n})^n \geq 1 - \frac{1}{e}$. Since payoffs are nonnegative, and since a rational receiver selects a $\mathbf{HIGH}$ action when one is available, the sender's overall expected utility is at least a $1 - \frac{1}{e}$ fraction of the optimal value of LP (5.3).

A.2 Proof of Theorem 5.2.5

This section is devoted to proving Theorem 5.2.5. Our proof starts from the ideas of (Gopalan et al., 2015), who show the #P-hardness for revenue or welfare maximization in several mechanism design problems. In one case, (Gopalan et al., 2015) reduce from the $\#P$-hard problem of computing the *Khintchine constant* of a vector. Our reduction also starts from this problem, but is much more involved: First, we exhibit a polytope which we term *Khintchine polytope*, and show that computing the Khintchine constant reduces to linear optimization over the Khintchine polytope. Second, we present a reduction from the membership problem for the Khintchine polytope to the computation of optimal sender utility in a particularly-crafted instance of persuasion with independent actions. Invoking the polynomial-time equivalence between membership checking and optimization (see, e.g., (Grötschel et al., 1988)), we conclude the #P-hardness of our problem. The main technical challenge we overcome is in the second step of our proof: given a point x which may or may not be in the Khintchine polytope $\mathcal{K}$, we construct a persuasion instance and a threshold T so that points in $\mathcal{K}$ encode signaling schemes, and the optimal sender utility is at least T if and only if $x \in \mathcal{K}$ and the scheme corresponding to x results in sender utility T.

The Khintchine Polytope

We start by defining the *Khintchine problem*, which is shown to be #P-hard in (Gopalan et al., 2015).

Definition 7. *(Khintchine Problem) Given a vector $a \in \mathbb{R}^n$, compute the* Khintchine constant *$K(a)$ of a, defined as follows:*

$$K(a) = \mathop{\mathbf{E}}_{\theta \sim \{\pm 1\}^n} [|\theta \cdot a|],$$

where θ is drawn uniformly *at random from $\{\pm 1\}^n$.*

To relate the Khintchine problem to Bayesian persuasion, we begin with a persuasion instance with n *i.i.d.* actions. Moreover, there are only two action types,[1] which we refer to as *type -1* and *type +1*. The state of nature is a uniform random draw from the set $\{\pm 1\}^n$, with the ith entry specifying the type of action i. It is easy to see that these actions are *i.i.d.*, with marginal probability $\frac{1}{2}$ for each type. We call this instance the *Khintchine-like* persuasion setting. As in Section 5.2.2 , we still use the *signature* to capture the payoff-relevant features of a signaling scheme. A signature for the Khintchine-like persuasion problem is of the form $\mathcal{M} = (M^1, ..., M^n)$ where $M^i \in \mathbb{R}^{n \times 2}$ for any $i \in [n]$. We pay special attention to signaling

[1] Recall from Section 5.2.2 that each type is associated with a pair (ξ, ρ), where ξ [ρ] is the payoff to the sender [receiver] if the receiver takes an action of that type.

$$\begin{array}{ll} \text{maximize} & \sum_{i=1}^{n} a_i(M^+_{i,+1} - M^+_{i,-1}) - \sum_{i=1}^{n} a_i(M^-_{i,+1} - M^-_{i,-1}) \\ \text{subject to} & (M^+, M^-) \in \mathcal{K}(n) \end{array} \tag{A.2}$$

Linear program for computing the Khintchine constant $K(a)$ for $a \in \mathbb{R}^n$

schemes which use only *two* signals, in which case we represent them using a *two-signal signature* of the form $(M^1, M^2) \in \mathbb{R}^{n\times 2} \times \mathbb{R}^{n\times 2}$. Recall that such a signature is *realizable* if there is a signaling scheme which uses only two signals, with the property that M^i_{jt} is the joint probability of the ith signal and the event that action j has type t. We now define the *Khintchine polytope*, consisting of a convex family of two-signal signatures.

Definition 8. *The* Khintchine polytope *is the family $\mathcal{K}(n)$ of* realizable *two-signal signatures (M^1, M^2) for the Khintchine-like persuasion setting which satisfy the additional constraints $M^1_{i,1} + M^1_{i,2} = \frac{1}{2}\, \forall i \in [n]$.*

We sometimes use $\mathcal{K}$ to denote the Khintchine polytope $\mathcal{K}(n)$ when the dimension n is clear from the context. Note that the constraints $M^1_{i,1} + M^1_{i,2} = \frac{1}{2}$, $\forall i \in [n]$ state that the first signal should be sent with probability $\frac{1}{2}$ (hence also the second signal). We now show that optimizing over the Khintchine polytope is $\#P$-hard by reducing the Kintchine problem to Linear program (A.2).

Lemma 25. *General linear optimization over the Khintchine polytope $\mathcal{K}$ is $\#P$-hard.*

Proof. For any given $a \in \mathbb{R}^n$, we reduce the computation of $K(a)$ – the Khintchine constant for a – to a linear optimization problem over the Khintchine polytope $\mathcal{K}$. Since our reduction will use two signals σ_+ and σ_- which correspond to the sign of $\theta \cdot a$, we will use (M^+, M^-) to denote the two matrices in the signature in lieu of $(M^1,\, M^2)$. Moreover, we use the two action types $+1$ and -1 to index the columns of each matrix. For example, $M^+_{i,-1}$ is the joint probability of signal σ_+ and the event that the ith action has type -1.

We claim that the Kintchine constant $K(a)$ equals the optimal objective value of the implicitly-described linear program (A.2). We denote this optimal objective value by $OPT(LP\text{ (A.2)})$. We first prove that $K(a) \leq OPT(LP\text{ (A.2)})$. Consider a signaling scheme φ in the Kintchine-like persuasion setting which simply outputs $\sigma_{sign(\theta\cdot a)}$ for each state of nature $\theta \in \{\pm 1\}^n$ (breaking tie uniformly at random if $\theta \cdot a = 0$). Since θ is drawn uniformly from $\{\pm 1\}^n$ and $sign(\theta \cdot a) = -sign(-\theta \cdot a)$, this scheme outputs each of the signals σ_- and σ_+ with probability $\frac{1}{2}$. Consequently, the two-signal signature of φ is a point in $\mathcal{K}$. Moreover,

evaluating the objective function of LP (A.2) on the two-signal signature (M^+, M^-) of φ yields $K(a) = \mathbf{E}_\theta[|\theta \cdot a|]$, as shown below.

$$\begin{aligned}
\mathop{\mathbf{E}}_{\theta}[|\theta \cdot a|] &= \mathop{\mathbf{E}}_{\theta}[\theta \cdot a|\sigma_+] \cdot \mathbf{Pr}(\sigma_+) + \mathop{\mathbf{E}}_{\theta}[-\theta \cdot a|\sigma_-] \cdot \mathbf{Pr}(\sigma_-) \\
&= \sum_{i=1}^{n} a_i \mathop{\mathbf{E}}_{\theta}[\theta_i|\sigma_+] \cdot \mathbf{Pr}(\sigma_+) - \sum_{i=1}^{n} a_i \mathop{\mathbf{E}}_{\theta}[\theta_i|\sigma_-] \times \mathbf{Pr}(\sigma_-) \\
&= \sum_{i=1}^{n} \Big(a_i[\mathbf{Pr}(\theta_i = 1|\sigma_+) - \mathbf{Pr}(\theta_i = -1|\sigma_+)] \cdot \mathbf{Pr}(\sigma_+) \Big) \\
&\quad - \sum_{i=1}^{n} \Big(a_i[\mathbf{Pr}(\theta_i = 1|\sigma_-) - \mathbf{Pr}(\theta_i = -1|\sigma_-)] \cdot \mathbf{Pr}(\sigma_-) \Big) \\
&= \sum_{i=1}^{n} \Big(a_i[\mathbf{Pr}(\theta_i = 1, \sigma_+) - \mathbf{Pr}(\theta_i = -1, \sigma_+)] \Big) - \sum_{i=1}^{n} \Big(a_i[\mathbf{Pr}(\theta_i = 1, \sigma_-) - \mathbf{Pr}(\theta_i = -1, \sigma_-)] \Big) \\
&= \sum_{i=1}^{n} a_i[M^+_{i,+1} - M^+_{i,-1}] - \sum_{i=1}^{n} a_i[M^-_{i,+1} - M^-_{i,-1}]
\end{aligned}$$

This concludes the proof that $K(a) \leq OPT(LP\text{ (A.2)})$.

Now we prove $K(a) \geq OPT(LP\text{ (A.2)})$. Take *any* signaling scheme which uses only two signals σ_+ and σ_-, and let (M^+, M^-) be its two-signal signature. Notice, however, that σ_+ now is only the "name" of the signal, and does not imply that $\theta \cdot a$ is positive. Nevertheless, it is still valid to reverse the above derivation until we reach

$$\sum_{i=1}^{n} a_i[M^+_{i,+1} - M^+_{i,-1}] - \sum_{i=1}^{n} a_i[M^-_{i,+1} - M^-_{i,-1}] \quad = \quad \mathop{\mathbf{E}}_{\theta}[\theta \cdot a|\sigma_+] \cdot \mathbf{Pr}(\sigma_+) + \mathop{\mathbf{E}}_{\theta}[-\theta \cdot a|\sigma_-] \cdot \mathbf{Pr}(\sigma_-).$$

Since $\theta \cdot a$ and $-\theta \cdot a$ are each no greater than $|\theta \cdot a|$, we have

$$\begin{aligned}
\mathop{\mathbf{E}}_{\theta}[\theta \cdot a|\sigma_+] \cdot \mathbf{Pr}(\sigma_+) + \mathop{\mathbf{E}}_{\theta}[-\theta \cdot a|\sigma_-] \cdot \mathbf{Pr}(\sigma_-) &\leq \mathop{\mathbf{E}}_{\theta}[|\theta \cdot a| \mid \sigma_+] \cdot \mathbf{Pr}(\sigma_+) + \mathop{\mathbf{E}}_{\theta}[|\theta \cdot a| \mid \sigma_-] \cdot \mathbf{Pr}(\sigma_-) \\
&= \mathop{\mathbf{E}}_{\theta}[|\theta \cdot a|] = K(a).
\end{aligned}$$

That is, the objective value of LP (A.2) is upper bounded by $K(a)$, as needed. □

Before we proceed to present the reduction from the membership problem for $\mathcal{K}$ to optimal persuasion, we point out an interesting corollary of Lemma 25.

Corollary 2. *Let $\mathcal{P}$ be the polytope of realizable signatures for a persuasion problem with n* i.i.d. *actions and m types (see Section 5.2.2). Linear optimization over $\mathcal{P}$ is $\#P$-hard, and this holds even when $m = 2$.*

Proof. Consider the Khintchine-like persuasion setting. It is easy to see that the Khintchine polytope $\mathcal{K}$ can be obtained from $\mathcal{P}$ by adding the constraints $M^{\sigma_i} = 0$ for $i \geq 3$ and $M^{\sigma_1}_{i,1} +$

$M^{\sigma_1}_{i,2} = \frac{1}{2}$ for $i \in [n]$, followed by a simple projection. Therefore, the membership problem for $\mathcal{K}$ can be reduced in polynomial time to the membership problem for $\mathcal{P}$, since the additional linear constraints can be explicitly checked in polynomial time. By the polynomial-time equivalence between optimization and membership, it follows that general linear optimization over $\mathcal{P}$ is $\#P$-hard. □

Remark A.2.1. *It is interesting to compare Corollary 2 to single item auctions with i.i.d. bidders, where the problem does admit a polynomial-time separation oracle for the polytope of realizable signatures via Border's Theorem (Border, 1991, 2007) and its algorithmic properties (Cai et al., 2012; Alaei et al., 2012). In contrast, the polytope of realizable signatures for Bayesian persuasion is #P-hard to optimize over. Nevertheless, in Section 5.2.2 we were indeed able to compute the optimal signaling scheme and sender utility for persuasion with i.i.d. actions. Corollary 2 conveys that it was crucial for our algorithm to exploit the special structure of the persuasion objective and the symmetry of the optimal scheme, since optimizing a general objective over $\mathcal{P}$ is #P-hard.*

Reduction

We now present a reduction from the membership problem for the Khintchine polytope to the computation of optimal sender utility for persuasion with independent actions. As the output of our reduction, we construct a persuasion instance of the following form. There are $n + 1$ actions. Action 0 is *special* – it deterministically results in sender utility ϵ and receiver utility 0. Here, we think of $\epsilon > 0$ as being small enough for our arguments to go through. The other n actions are *regular*. Action $i > 0$ *independently* results in sender utility $-a_i$ and receiver utility a_i with probability $\frac{1}{2}$ (call this the type 1_i), or sender utility $-b_i$ and receiver utility b_i with probability $\frac{1}{2}$ (call this the type 2_i). Note that the sender and receiver utilities are *zero-sum* for both types. Notice that, though each regular action's type distribution is uniform over its two types, the actions here are *not* identical because the associated payoffs — specified by a_i and b_i for each action i — are different for different actions. Since the special action is deterministic and the probability of its (only) type is 1 in any signal, we can interpret any $(M^1, M^2) \in \mathcal{K}(n)$ as a two-signal signature for our persuasion instance (the row corresponding to the special action 0 is implied). For example, $M^1_{i,2}$ is the joint probability of the first signal and the event that action i has type 2_i. Our goal is to reduce membership checking for $\mathcal{K}(n)$ to computing the optimal expected sender utility for a persuasion instance with carefully chosen parameters $\{a_i\}_{i=1}^n$, $\{b_i\}_{i=1}^n$, and ϵ.

In relating optimal persuasion to the Khintchine polytope, there are two main difficulties: (1) $\mathcal{K}$ consists of two-signal signatures, so there should be an optimal scheme to our persuasion instance which uses only two signals; (2) To be consistent with the definition of $\mathcal{K}$, such an

optimal scheme should send each signal with probability exactly $\frac{1}{2}$. We will design specific ϵ, a_i, b_i to accomplish both goals.

For notational convenience, we will again use (M^+, M^-) to denote a typical element in $\mathcal{K}$ instead of (M^1, M^2) because, as we will see later, the two constructed signals will induce positive and negative sender utilities, respectively. Notice that there are only n degrees of freedom in $(M^+, M^-) \in \mathcal{K}$. This is because $M^+ + M^-$ is the all-$\frac{1}{2}$ matrix in $\mathbb{R}^{n\times 2}$, corresponding to the prior distribution of states of nature (by the definition of realizable signatures). Moreover, $M^+_{i,1} + M^-_{i,2} = \frac{1}{2}$ for all $i \in [n]$ (by the definition of $\mathcal{K}$). Therefore, we must have

$$M^+_{i,1} = M^-_{i,2} = \frac{1}{2} - M^+_{i,2} = \frac{1}{2} - M^-_{i,1}.$$

This implies that we can parametrize signatures $(M^+, M^-) \in \mathcal{K}$ by a vector $\boldsymbol{x} \in [0, \frac{1}{2}]^n$, where $M^+_{i,1} = M^-_{i,2} = x_i$ and $M^+_{i,2} = M^-_{i,1} = \frac{1}{2} - x_i$ for each $i \in [n]$. For any $\boldsymbol{x} \in [0, \frac{1}{2}]^n$, let $\mathcal{M}(\boldsymbol{x})$ denote the signature (M^+, M^-) defined by $\boldsymbol{x}$ as just described.

We can now restate the membership problem for $\mathcal{K}$ as follows: given $\boldsymbol{x} \in [0, \frac{1}{2}]^n$, determine whether $\mathcal{M}(\boldsymbol{x}) \in \mathcal{K}$. When any of the entries of $\boldsymbol{x}$ equals 0 or $\frac{1}{2}$ this problem is trivial,[2] so we assume without loss of generality that $\boldsymbol{x} \in (0, \frac{1}{2})^n$. Moreover, when $x_i = \frac{1}{4}$ for some i, it is easy to see that a signaling scheme with signature $\mathcal{M}(\boldsymbol{x})$, if one exists, must choose its signal independently of the type of action i, and therefore $\mathcal{M}(\boldsymbol{x}) \in \mathcal{K}(n)$ if and only if $\mathcal{M}(\boldsymbol{x_{-i}}) \in \mathcal{K}(n-1)$. This allows us to assume without loss of generality that $x_i \neq \frac{1}{4}$ for all i.

Given $\boldsymbol{x} \in (0, \frac{1}{2})^n$ with $x_i \neq \frac{1}{4}$ for all i, we construct specific ϵ and a_i, b_i for all i such that we can determine whether $\mathcal{M}(\boldsymbol{x}) \in \mathcal{K}$ by simply looking at the optimal sender utility in the corresponding persuasion instance. We choose parameters a_i and b_i to satisfy the following two equations.

$$x_i a_i + (\frac{1}{2} - x_i) b_i = 0. \tag{A.3}$$

$$(\frac{1}{2} - x_i) a_i + x_i b_i = \frac{1}{2}. \tag{A.4}$$

We note that the above linear system always has a solution when $x_i \neq \frac{1}{4}$, which we assumed previously. We make two observations about our choice of a_i and b_i. First, the *prior* expected receiver utility $\frac{1}{2}(a_i + b_i)$ equals $\frac{1}{2}$ for all actions i (by simply adding Equation (A.3) and (A.4)). Second, a_i and b_i are both non-zero, and this follows easily from our assumption that $x_i \in (0, \frac{1}{2})$.

Now we show how to determine whether $\mathcal{M}(\boldsymbol{x}) \in \mathcal{K}$ by only examining the optimal sender utility in the constructed persuasion instance. We start by showing that restricting to two-signal schemes is without loss of generality in our instance.

[2] If x_i is 0 or $\frac{1}{2}$, then $\mathcal{M}(x) \in \mathcal{K}$ if and only if $x_j = \frac{1}{4}$ for all $j \neq i$. This is because the corresponding signaling scheme must choose its signal based solely on the type of action i.

Lemma 26. *There exists an optimal persuasive signaling scheme which uses at most* two *signals: one signal recommends the special action, and the other recommends some regular action.*

Proof. Recall that an optimal persuasive scheme uses $n+1$ signals, with signal σ_i recommending action i for $i = 0, 1, ..., n$. Fix such a scheme, and let α_i denote the probability of signal σ_i. Signal σ_i induces posterior expected receiver utility $r_j(\sigma_i)$ and sender utility $s_j(\sigma_i)$ for each action j. For a regular action $j \neq 0$, we have $s_j(\sigma_i) = -r_j(\sigma_i)$ for all i due to the zero-sum nature of our construction. Notice that $r_i(\sigma_i) \geq 0$ for all regular actions $i \neq 0$, since otherwise the receiver would prefer action 0 over action i. Consequently, for each signal σ_i with $i \neq 0$, the receiver derives non-negative utility and the sender derives non-positive utility.

We claim that merging signals $\sigma_1, \sigma_2, \ldots, \sigma_n$ — i.e., modifying the signaling scheme to output the same signal σ^* in lieu of each of them — would not decrease the sender's expected utility. Recall that persuasiveness implies that $r_i(\sigma_i) = \max_{j=0}^{n} r_j(\sigma_i)$. Using Jensen's inequality, we get

$$\sum_{i=1}^{n} \alpha_i r_i(\sigma_i) \geq \max_{j=0}^{n} \left[\sum_{i=1}^{n} \alpha_i r_j(\sigma_i) \right]. \tag{A.5}$$

If the maximum in the right hand side expression of (A.5) is attained at $j^* = 0$, the receiver will choose the special action 0 when presented with the merged signal σ^*. Recalling that $s_i(\sigma_i)$ is non-positive for $i \neq 0$, this can only improve the sender's expected utility. Otherwise, the receiver chooses a regular action $j^* \neq 0$ when presented with σ^*, resulting in a total contribution of $\sum_{i=1}^{n} \alpha_i r_{j^*}(\sigma_i)$ to the receiver's expected utility from the merged signal, down from the total contribution of $\sum_{i=1}^{n} \alpha_i r_i(\sigma_i)$ by the original signals $\sigma_1, \ldots, \sigma_n$. Recalling the zero-sum nature of our construction for regular actions, the merged signal σ^* contributes $\sum_{i=1}^{n} \alpha_i s_{j^*}(\sigma_i) = -\sum_{i=1}^{n} \alpha_i r_{j^*}(\sigma_i)$ to the sender's expected utility, up from a total contribution of $\sum_{i=1}^{n} \alpha_i s_i(\sigma_i) = -\sum_{i=1}^{n} \alpha_i r_i(\sigma_i)$ by the original signals $\sigma_1, \ldots, \sigma_n$. Therefore, the sender is not worse off by merging the signals. Moreover, interpreting σ^* as a recommendation for action j^* yields persuasiveness. □

Therefore, in characterizing the optimal solution to our constructed persuasion instance, it suffices to analyze two-signal schemes of the the form guaranteed by Lemma 26. For such a scheme, we denote the signal that recommends the special action 0 by σ_+ (indicating that the sender derives positive utility ϵ), and denote the other signal by σ_- (indicating that the sender derives negative utility, as we will show). For convenience, in the following discussion we use the expression "payoff from a signal" to signify the expected payoff of a player conditioned on that signal multiplied by the probability of that signal. For example, the *sender's expected payoff from signal* σ_- equals the sender's expected payoff conditioned on signal σ_- multiplied by the overall probability that the scheme outputs σ_-, assuming the receiver follows the scheme's (persuasive)

recommendations. We also use the expression "payoff from an action in a signal" to signify the posterior expected payoff of a player for that action conditioned on the signal, multiplied by the probability that the scheme outputs the signal. For example, the *receiver's expected payoff from action i in signal σ_+* equals $\alpha_+ \cdot r_i(\sigma_+)$, where $r_i(\sigma_+)$ is the receiver's posterior expected payoff from action i given signal σ_+, and α_+ is the overall probability of signal σ_+.

Lemma 27. *Fix a persuasive scheme with signals σ_- and σ_+ as described above. The sender's expected payoff from signal σ_- is at most $-\frac{1}{2}$. Moreover, if the sender' expected payoff from σ_- is exactly $-\frac{1}{2}$, then for each regular action i the expected payoff of both the sender and the receiver from action i in signal σ_+ equals 0.*

Proof. Assume that signal σ_+ [σ_-] is sent with probability α_+ [α_-] and induces posterior expected receiver payoff $r_i(\sigma_+)$ [$r_i(\sigma_-)$] for each action i. Recall from our construction that the *prior* expected payoff of each regular action $i \neq 0$ equals $\frac{1}{2}a_i + \frac{1}{2}b_i = \frac{1}{2}$. Since the prior expectation must equal the expected posterior expectation, it follows that $\alpha_+ \cdot r_i(\sigma_+) + \alpha_- \cdot r_i(\sigma_-) = \frac{1}{2}$ when i is regular. The receiver's reward from the special action is deterministically 0, and therefore persuasiveness implies that $r_i(\sigma_+) \leq 0$ for each regular action i. It follows that $\alpha_- \cdot r_i(\sigma_-) = \frac{1}{2} - \alpha_+ \cdot r_i(\sigma_+) \geq \frac{1}{2}$ for regular actions i. In other words, the receiver's expected payoff from each regular action in signal σ_- is at least $\frac{1}{2}$. By the zero-sum nature of our construction, the sender's expected payoff from each regular action in signal σ_- is at most $-\frac{1}{2}$. Since σ_- recommends a regular action, we conclude that the sender's expected payoff from σ_- is at most $-\frac{1}{2}$.

Now assume that the sender's expected payoff from σ_- is exactly $-\frac{1}{2}$. By the zero-sum property, persuasiveness, and the above-established fact that $\alpha_- \cdot r_i(\sigma_-) \geq \frac{1}{2}$ for regular actions i, it follows that the receiver's expected payoff from each regular action in signal σ_- is *exactly* $\frac{1}{2}$. Recalling that $\alpha_+ \cdot r_i(\sigma_+) + \alpha_- \cdot r_i(\sigma_-) = \frac{1}{2}$ when i is regular, we conclude that the receiver's expected payoff from a regular action in signal σ_+ equals 0. By the zero-sum property for regular actions, the same is true for the sender.

□

The key to the remainder of our reduction is to choose a small enough value for the parameter ϵ — the sender's utility from the special action — so that the optimal signaling scheme satisfies the property mentioned in Lemma 27: The sender's expected payoff from signal σ_- is exactly equal to its maximum possible value of $-\frac{1}{2}$. In other words, we must make ϵ so small so that the sender prefers to not sacrifice *any* of her payoff from σ_- in order to gain utility from the special action recommended by σ_+. Notice that this upper bound of $-\frac{1}{2}$ is indeed achievable: the uninformative signaling scheme which recommends an arbitrary regular action has this property. We now show that a "small enough" ϵ indeed exists. The key idea behind this existence proof is

the following: We start with a signaling scheme which maximizes the sender's payoff from σ_- at $-\frac{1}{2}$, and moreover corresponds to a vertex of the polytope of persuasive signatures. When $\epsilon > 0$ is smaller than the "bit complexity" of the vertices of this polytope, moving to a different vertex — one with lower sender payoff from σ_- — will result in more utility loss from σ_- than utility gain from σ_+. We show that $\epsilon > 0$ with polynomially many bits suffices, and can be computed in polynomial time.

Let $\mathcal{P}_2$ be the family of all *realizable* two-signal signatures (again, ignoring action 0). It is easy to see that $\mathcal{P}_2$ is a polytope, and importantly, all entries of any vertex of $\mathcal{P}_2$ are integer multiples of $\frac{1}{2^n}$. This is because every vertex of $\mathcal{P}_2$ corresponds to a deterministic signaling scheme which partitions the set of states of nature, and every state of nature occurs with probability $1/2^n$. As a result, all vertices of $\mathcal{P}_2$ have $\mathcal{O}(n)$ bit complexity.

To ease our discussion, we use a compact representation for points in $\mathcal{P}_2$. In particular, any point in $\mathcal{P}_2$ can be captured by $n+1$ variables: variable p denotes the probability of sending signal σ_+, and variable y_i denotes the joint probability of signal σ_+ and the event that action i has type 1_i. It follows that joint probability of type 2_i and signal σ_+ is $p - y_i$, and the probabilities associated with signal σ_- are determined by the constraint that $M^+ + M^-$ is the all-$\frac{1}{2}$ matrix. With some abuse of notation, we use $\mathcal{M}(p, \boldsymbol{y}) = (M^+, M^-)$ to denote the signature in $\mathcal{P}_2$ corresponding to the probability p and n-dimensional vector $\boldsymbol{y}$. Now we consider the following two linear programs.

$$\begin{aligned}
\text{maximize} \quad & p\epsilon + u \\
\text{subject to} \quad & \mathcal{M}(p, \boldsymbol{y}) \in \mathcal{P}_2 \\
& y_i a_i + (p - y_i) b_i \le 0, && \text{for } i = 1, \dots, n. \\
& u \le -[(\tfrac{1}{2} - y_i) a_i + (\tfrac{1}{2} - p + y_i) b_i], && \text{for } i = 1, \dots, n.
\end{aligned} \tag{A.6}$$

$$\begin{aligned}
\text{maximize} \quad & u \\
\text{subject to} \quad & \mathcal{M}(p, \boldsymbol{y}) \in \mathcal{P}_2 \\
& y_i a_i + (p - y_i) b_i \le 0, && \text{for } i = 1, \dots, n. \\
& u \le -[(\tfrac{1}{2} - y_i) a_i + (\tfrac{1}{2} - p + y_i) b_i], && \text{for } i = 1, \dots, n.
\end{aligned} \tag{A.7}$$

Linear programs (A.6) and (A.7) are identical except for the fact that the objective of LP (A.6) includes the additional term $p\epsilon$. LP (A.6) computes precisely the optimal expected sender utility in our constructed persuasion instance: The first set of inequality constraints are the persuasiveness constraints for the signal σ_+ recommending action 0; The second set of inequality constraints state that the sender's payoff from signal σ_- is the minimum among all actions, as implied by the zero-sum nature of our construction; The objective is the sum of the sender's payoffs from signals σ_+ and σ_-. Notice that the persuasiveness constraints for signal σ_-, namely $(\frac{1}{2} - y_i) a_i + (\frac{1}{2} - p + y_i) b_i \ge 0$ for all $i \neq 0$, are implicitly satisfied because $\frac{1}{2} a_i + \frac{1}{2} b_i = \frac{1}{2}$ by our

construction and $(\frac{1}{2}-y_i)a_i+(\frac{1}{2}-p+y_i)b_i = \frac{1}{2}a_i+\frac{1}{2}b_i-[y_ia_i+(p-y_i)b_i] \geq \frac{1}{2}-0 > 0$. On the other hand, LP (A.7) maximizes the sender's expected payoff from signal σ_-. Observe that the optimal objective value of LP (A.7) is precisely $-\frac{1}{2}$ because $u \leq -[(\frac{1}{2}-y_i)a_i+(\frac{1}{2}-p+y_i)b_i] \leq -\frac{1}{2}$ for all $i \neq 0$, and equality is attained, for example, at $p = 0$ and $\boldsymbol{y} = 0$.

Let $\widetilde{\mathcal{P}_2}$ be the set of all feasible $(u, \mathcal{M}(p, \boldsymbol{y}))$ for LP (A.6) (and LP (A.7)). Obviously, $\widetilde{\mathcal{P}_2}$ is a polytope. We now argue that all vertices of $\widetilde{\mathcal{P}_2}$ have bit complexity polynomial in n and the bit complexity of $\boldsymbol{x} \in (0, \frac{1}{2})^n$. In particular, denote the bit complexity of $\boldsymbol{x}$ by ℓ. Since a_i, b_i are computed by a two-variable two-equation linear system involving x_i (Equations (A.3) and (A.4)), they each have $O(\ell)$ bit complexity. Consequently, all the explicitly described facets of $\widetilde{\mathcal{P}_2}$ have $\mathcal{O}(\ell)$ bit complexity. Moreover, since each vertex of $\mathcal{P}_2$ has $\mathcal{O}(n)$ bit complexity, each facet of $\mathcal{P}_2$ then has $\mathcal{O}(n^3)$ bit complexity, i.e., the coefficients of inequalities that determine the facets have $\mathcal{O}(n^3)$ bit complexity. This is due to the fact that facet complexity of a rational polytope is upper bounded by a *cubic* polynomial of the vertex complexity and *vice versa* (see, e.g., (Schrijver, 2003)). To sum up, any facet of polytope $\widetilde{\mathcal{P}_2}$ has bit complexity $\mathcal{O}(n^3 + \ell)$, and therefore any vertex of $\widetilde{\mathcal{P}_2}$ has $\mathcal{O}(n^9\ell^3)$ bit complexity.

Let the polynomial $B(n, \ell) = O(n^9\ell^3)$ be an upper bound on the maximum bit complexity of vertices of $\widetilde{\mathcal{P}_2}$. Now we are ready to set the value of ϵ. LP (A.6) always has an optimal vertex solution which we denote as $(u^*, \mathcal{M}^*)$. Recall that $u \leq -\frac{1}{2}$ for all points $(u, \mathcal{M}(p, \boldsymbol{y}))$ in $\widetilde{\mathcal{P}_2}$ and $u = -\frac{1}{2}$ is attainable at some vertices. Since all vertices of $\widetilde{\mathcal{P}_2}$ have $B(n, \ell)$ bit complexity, (u^*, M^*) must either satisfy either $u^* = -\frac{1}{2}$ or $u^* \leq -\frac{1}{2} - 2^{-B(n,\ell)}$. Therefore, it suffices to set $\epsilon = 2^{-n \cdot B(n,\ell)}$, which is a number with polynomial bit complexity. As a result, any optimal vertex solution to LP (A.6) must satisfy $u^* = -\frac{1}{2}$, since the loss incurred by moving to any other vertex with $u < -\frac{1}{2}$ can never be compensated for by the other term $p\epsilon < \epsilon$.

With such a small value of ϵ, the sender's goal is to send signal σ_+ with probability as high as possible, subject to the constraint that her utility from σ_- is precisely $-\frac{1}{2}$. In other words, signal σ_+ must induce expected receiver/sender utility precisely 0 for each regular action $i \neq 0$ (see Lemma 27). This characterization of the optimal scheme now allows us to determine whether $\mathcal{M}(\boldsymbol{x}) \in \mathcal{K}$ by inspecting the sender's optimal expected utility. The following Lemma completes our proof of Theorem 5.2.5.

Lemma 28. *Given the small enough value of ϵ described above, the sender's expected utility in the optimal signaling scheme for our constructed persuasion instance is at least $\frac{1}{2}(\epsilon - 1)$* if and only if $\mathcal{M}(\boldsymbol{x}) \in \mathcal{K}$.

Proof. $\Leftarrow$: If $\mathcal{M}(\boldsymbol{x}) \in \mathcal{K}$, then by our choice of a_i, b_i (recall Equations (A.3) and (A.4)), the signaling scheme implementing $\mathcal{M}(\boldsymbol{x})$ is persuasive, the sender's payoff from signal σ_+ is $\frac{1}{2}\epsilon$, and her payoff from σ_- is $-\frac{1}{2}$. Therefore, the optimal sender utility is at least $\frac{1}{2}\epsilon - \frac{1}{2}$.

$\Rightarrow$: Let $\mathcal{M}(p, \boldsymbol{y})$ be the signature of a vertex optimal signaling scheme in LP (A.6). By our choice of ϵ we know that the sender payoff from signal σ_- must be exactly $-\frac{1}{2}$. Therefore, to achieve overall sender utility at least $\frac{1}{2}\epsilon - \frac{1}{2}$, signal σ_+ must be sent with probability $p \geq \frac{1}{2}$, and the receiver's payoff from each regular action $i \neq 0$ in signal σ_+ is exactly 0. That is, $y_i a_i + (p - y_i)b_i = 0$. By construction, we also have that $x_i a_i + (0.5 - x_i)b_i = 0$ and $a_i, b_i \neq 0$, which imply that $\frac{y_i}{x_i} = \frac{p - y_i}{0.5 - x_i}$ and, furthermore, that $y_i \geq x_i$ since $p \geq \frac{1}{2}$. Now let φ be a signaling scheme with the signature $\mathcal{M}(p, \boldsymbol{y})$. We can post-process φ so it has signature $\mathcal{M}(\boldsymbol{x})$ as follows: whenever φ outputs the signal σ_+, flip a biased random coin to output σ_+ with probability $\frac{0.5}{p}$ and output σ_- otherwise. By using the identity $\frac{y_i}{x_i} = \frac{p - y_i}{0.5 - x_i}$, it is easy to see that this adjusted signaling scheme has signature $\mathcal{M}(\boldsymbol{x})$. $\square$

A.3 Omitted Proofs from Section 5.2.4

A.3.1 A Bicriteria FPTAS

Proof of Lemma 4

Fix ϵ, K, and λ, and let φ denote the resulting signaling scheme implemented by Algorithm 2. Let $\theta \sim \lambda$ denote the input to φ, and $\sigma \sim \varphi(\theta)$ denote its output. First, we condition on the empirical sample $\widetilde{\lambda} = \{\theta_1, \ldots, \theta_K\}$ without conditioning on the index ℓ of the input state of nature θ, and show that ϵ-persuasiveness holds subject to this conditioning. The principle of deferred decisions implies that, subject to this conditioning, θ is uniformly distributed in $\widetilde{\lambda}$. By definition of linear program (5.4), the signaling scheme $\widetilde{\varphi}$ computed in Step 3 is ϵ-persuasive scheme for the empirical distribution $\widetilde{\lambda}$. Since $\sigma \sim \widetilde{\varphi}(\theta)$ and θ is conditionally distributed according to $\widetilde{\lambda}$, this implies that all ϵ-persuasiveness constraints conditionally hold; formally, the following holds for each pair of actions i and j:

$$\mathbf{E}[r_i(\theta)|\sigma = \sigma_i, \widetilde{\lambda}] \geq \mathbf{E}[r_j(\theta)|\sigma = \sigma_i, \widetilde{\lambda}] - \epsilon$$

Removing the conditioning on $\widetilde{\lambda}$ and invoking linearity of expectations shows that φ is ϵ-persuasive for λ, completing the proof.

Proof of Lemma 5

As in the proof of Lemma 4, we condition on the empirical sample $\widetilde{\lambda} = \{\theta_1, \ldots, \theta_K\}$ and observe that θ is uniformly distributed in $\widetilde{\lambda}$ after this conditioning. The conditional expectation of sender utility then equals $\sum_{k=1}^{K} \sum_{i=1}^{n} \frac{1}{K} \widetilde{\varphi}(\theta_k, \sigma_i) s_i(\theta_k)$, where $\widetilde{\varphi}$ is the signaling scheme computed in Step 3 based on $\widetilde{\lambda}$. Since this is precisely the optimal value of the LP (5.4) solved in Step 3, removing the conditioning and invoking linearity of expectations completes the proof.

Proof of Lemma 6

Recall that linear program (5.1) solves for the optimal persuasive scheme for λ. It is easy to see that the linear program (5.4) solved in step 3 is simply the instantiation of LP (5.1) for the empirical distribution $\widetilde{\lambda}$ consisting of K samples from λ. To prove the lemma, it would suffice to show that the optimal persuasive scheme φ^* corresponding to LP (5.1) remains ϵ-persuasive and ϵ-optimal for the distribution $\widetilde{\lambda}$, with high probability. Unfortunately, this approach fails because polynomially-many samples from λ are not sufficient to approximately preserve the persuasiveness constraints corresponding to low-probability signals (i.e., signals which are output with probability smaller than inverse polynomial in n). Nevertheless, we show in Claim 4 that there exists an approximately optimal solution $\widehat{\varphi}$ to LP (5.1) with the property that every signal

σ_i is either *large*, which we define as being output by $\widehat{\varphi}$ with probability at least $\frac{\epsilon}{4n}$ assuming $\theta \sim \lambda$, or *honest* in that only states of nature θ with $i \in \operatorname{argmax}_j r_j(\theta)$ are mapped to it. It is easy to see that sampling preserves persuasiveness exactly for honest signals. As for large signals, we employ tail bounds and the union bound to show that polynomially many samples suffice to approximately preserve persuasiveness (Claim 5).

Claim 4. *There is a signaling scheme $\widehat{\varphi}$ which is persuasive for λ, induces sender utility $u_s(\widehat{\varphi}, \lambda) \geq OPT - \frac{\epsilon}{2}$ on λ, and such that every signal of $\widehat{\varphi}$ is either large or honest.*

Proof. Let φ^* be the optimal persuasive scheme for λ — i.e. the optimal solution to LP (5.1). We call a signal σ *small* if it is output by φ^* with probability less than $\frac{\epsilon}{4n}$, i.e. if $\sum_{\theta \in \Theta} \lambda_\theta \varphi^*(\theta, \sigma) < \frac{\epsilon}{4n}$, and otherwise we call it *large*. Let $\widehat{\varphi}$ be the scheme which is defined as follows: on input θ, it first samples $\sigma \sim \varphi^*(\theta)$; if σ is large then $\widehat{\varphi}$ simply outputs σ, and otherwise it recommends an action maximizing receiver utility in state of nature θ —- i.e., outputs $\sigma_{i'}$ for $i' \in \operatorname{argmax}_i r_i(\theta)$. It is easy to see that every signal of $\widehat{\varphi}$ is either large or honest. Moreover, since φ^* is persuasive and $\widehat{\varphi}$ only replaces recommendations of φ^* with "honest" recommendations, it is easy to check that $\widehat{\varphi}$ is persuasive for λ. Finally, since the total probability of small signals in φ^* is at most $\frac{\epsilon}{4}$, and utilities are in $[-1, 1]$, the sender's expected utility from $\widehat{\varphi}$ is no worse than $\frac{\epsilon}{2}$ smaller than her expected utility from φ^*. □

Claim 5. *Let $\widehat{\varphi}$ be the signaling scheme from Claim 4. With probability at least $1 - \frac{\epsilon}{8}$ over the sample $\widetilde{\lambda}$, $\widehat{\varphi}$ is ϵ-persuasive for $\widetilde{\lambda}$, and moreover $u_s(\widehat{\varphi}, \widetilde{\lambda}) \geq u_s(\widehat{\varphi}, \lambda) - \frac{\epsilon}{4}$.*

Proof. Recall that $\widehat{\varphi}$ is persuasive for λ, and every signal is either large or honest. Since $\widetilde{\lambda}$ is a set of samples from λ, it is easy to see that persuasiveness constraints pertaining to the honest signals continue to hold over $\widetilde{\lambda}$. It remains to show that persuasiveness constraints for large signals, as well as expected sender utility, are approximately preserved when replacing λ with $\widetilde{\lambda}$.

Recall that persuasiveness requires that $\mathbf{E}_\theta[\widehat{\varphi}(\theta, \sigma_i)(r_i(\theta) - r_j(\theta))] \geq 0$ for each $i, j \in [n]$. Moreover, the sender's expected utility can be written as $\mathbf{E}_\theta[\sum_{i=1}^n \widehat{\varphi}(\theta, \sigma_i) s_i(\theta)]$. The left hand side of each persuasiveness constraint evaluates the expectation of a fixed function of θ with range $[-2, 2]$, whereas the sender's expected utility evaluates the expectation of a function of θ with range in $[-1, 1]$. Standard tail bounds and the union bound, coupled with our careful choice of the number of samples K, imply that replacing distribution λ with $\widetilde{\lambda}$ approximately preserves each of these $n^2 + 1$ quantities to within an additive error of $\frac{\epsilon^2}{4n}$ with probability at least $1 - \frac{\epsilon}{8}$. This bound on the additive loss translates to ϵ-persuasiveness for the large signals, and is less than the permitted decrease of $\frac{\epsilon}{4}$ for expected sender utility. □

The above claims, coupled with the fact that sender payoffs are bounded in $[-1, 1]$, imply that the expected optimal value of linear program (5.4) is at least $OPT - \epsilon$, as needed.

	Rainy	Sunny
Walk	$1-\delta$	1
Drive	1	0

Table A.1: Receiver's payoffs in rain and shine example

A.3.2 Information-Theoretic Barriers

Impossibility of Persuasiveness (Proof of Theorem 5.2.7 (a))

Consider a setting with two states of nature, which we will conveniently refer to as *rainy* and *sunny*. The receiver, who we may think of as a daily commuter, has two actions: *walk* and *drive*. The receiver slightly prefers driving on a rainy day, and strongly prefers walking on a sunny day. We summarize the receiver's payoff function, parametrized by $\delta > 0$, in Table A.1. The sender, who we will think of as a municipality with black-box sample access to weather reports drawn from the same distribution as the state of nature, strongly prefers that the receiver chooses walking regardless of whether it is sunny or rainy: we let $s_{walk} = 1$ and $s_{drive} = 0$ in both states of nature.

Let λ_r be the point distribution on the rainy state of nature, and let λ_s be such that $\mathbf{Pr}_{\lambda_s}[\text{rainy}] = \frac{1}{1+2\delta}$ and $\mathbf{Pr}_{\lambda_s}[\text{sunny}] = \frac{2\delta}{1+2\delta}$. It is easy to see that the unique direct persuasive scheme for λ_r always recommends driving, and hence results in expected sender utility of 0. In contrast, a simple calculation shows that always recommending walking is persuasive for λ_s, and results in expected sender utility 1. If algorithm $\mathcal{A}$ is persuasive and c-optimal for a constant $c < 1$, then $\mathcal{A}(\lambda_r)$ must never recommend walking whereas $\mathcal{A}(\lambda_s)$ must recommend walking with constant probability at least $(1 - c)$ overall (in expectation over the input state of nature $\theta \sim \lambda_s$ as well as all other internal randomness). Consequently, given a black box distribution $\mathcal{D} \in \{\lambda_r, \lambda_s\}$, evaluating $\mathcal{A}(\mathcal{D}, \theta)$ on a random draw $\theta \sim \mathcal{D}$ yields a tester which distinguishes between λ_r and λ_s with constant probability $1 - c$.

Since the total variation distance between λ_r and λ_s is $O(\delta)$, it is well known (and easy to check) that any black-box algorithm which distinguishes between the two distributions with $\Omega(1)$ success probability must take $\Omega(\frac{1}{\delta})$ samples in expectation when presented with one of these distributions. As a consequence, the average-case sample complexity of $\mathcal{A}$ on either of λ_r and λ_s is $\Omega(\frac{1}{\delta})$. Since $\delta > 0$ can be made arbitrarily small, this completes the proof.

Impossibility of Optimality (Proof of Theorem 5.2.7 (b))

Consider a setting with three actions $\{1, 2, 3\}$ and three corresponding states of nature $\theta_1, \theta_2, \theta_3$. In each state θ_i, the receiver derives utility 1 from action i and utility 0 from the other actions. The sender, on the other hand, derives utility 1 from action 3 and utility 0 from actions 1 and 2.

	$\mathbf{Pr}[\theta_1]$	$\mathbf{Pr}[\theta_2]$	$\mathbf{Pr}[\theta_3]$
λ	$1-2\delta$	2δ	0
λ'	$1-2\delta$	δ	δ

Table A.2: Two distributions on three actions

For an arbitrary parameter $\delta > 0$, we define two distributions λ and λ' over states of nature with total variation distance δ, illustrated in Table A.2.

Assume algorithm $\mathcal{A}$ is optimal and c-persuasive for a constant $c < \frac{1}{4}$. The optimal persuasive scheme for λ' results in expected sender utility 3δ by recommending action 3 whenever the state of nature is θ_2 or θ_3, and with probability $\frac{\delta}{1-2\delta}$ when the state of nature is θ_1. Some calculation reveals that in order to match this expected sender utility subject to c-persuasiveness, signaling scheme $\varphi' = \mathcal{A}(\lambda')$ must satisfy $\varphi'(\theta_2, \sigma_3) \geq \mu$ for $\mu = 1 - 4c > 0$. In other words, φ' must recommend action 3 a constant fraction of the time when given state θ_2 as input. In contrast, since $c < \frac{1}{2}$ it is easy to see that $\varphi = \mathcal{A}(\lambda)$ can never recommend action 3: for any signal, the posterior expected receiver reward for action 3 is 0, whereas one of the other two actions must have posterior expected receiver reward at least $\frac{1}{2}$. It follows that given $D \in \{\lambda, \lambda'\}$, a call to $\mathcal{A}(\mathcal{D}, \theta_2)$ yields a tester which distinguishes between λ and λ' with constant probability μ. Since λ and λ' have statistical distance δ, we conclude that the worst case sample complexity of $\mathcal{A}$ on either of λ or λ' is $\Omega(\frac{1}{\delta})$. Since $\delta > 0$ can be made arbitrarily small, this completes the proof.

Appendix B

Omissions From Section 6.2.3.1

B.1 Omitted Proofs

Proof of Lemma 12

The linear program for solving zero-sum SEGs can be written as follows, which is a slight modification to LP (6.6):

$$
\begin{array}{lll}
\max & u & \\
\text{s.t.} & u \le x_i U_+^d(i) + w_i U_-^d(i) + U_\sigma^d(\pi_i^+, \pi_i^-) & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_+} p_e = x_i & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_{s+}} p_e = y_i & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_{s-}} p_e = z_i & \forall\, i \in [n] \\
& x_i + y_i + z_i + w_i = 1 & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}} p_{\mathbf{e}} = 1 & \\
& p_{\mathbf{e}} \ge 0 & \forall\, \mathbf{e} \in \mathcal{E} \\
& U_\sigma^d(\pi_i^+, \pi_i^-) \le 0 & \forall\, i \in [n] \\
& (y_i - \pi_i^+) U_+^d(i) + (z_i - \pi_i^-) U_-^d(i) \ge 0 & \forall\, i \in [n] \\
& 0 \le \pi_i^+ \le y_i, \qquad 0 \le \pi_i^- \le z_i & \forall\, i \in [n]
\end{array}
\tag{B.1}
$$

We first prove a useful property of the optimal solution of LP (B.1). In particular, we show that there always exists an optimal solution to LP (B.1) that satisfies $\pi_i^- = z_i \,\forall i \in [n]$.

First, we claim that it is without loss of generality to assume that the optimal solution satisfied either $y_i = \pi_i^+$ or $z_i = \pi_i^-$. Otherwise, we can increase π_i^+ by $\frac{\epsilon}{U_+^d(i)}$ and π_i^- by $\frac{\epsilon}{-U_-^d(i)}$ without violating constraints and changing the objective value. Once one of the π_i^+, π_i^- reaches its upper bound, we have $y_i = \pi_i^+$ or $z_i = \pi_i^-$ and the solution remains optimal.

Now, if $\pi_i^- = z_i$, then we are done. If $\pi_i^+ = y_i$, we have $0 \le (y_i - \pi_i^+)U_+^d(i) + (z_i - \pi_i^-)U_-^d(i) = (z_i - \pi_i^-)U_-^d(i) \le 0$, which implies $z_i = \pi_i^-$ or $U_-^d(i) = 0$. In the later case, we can arbitrary set π_i^- to be z_i without affecting anything neither.

Therefore, adding the constraint $z_i = \pi_i^-$ will not affect the optimal value of linear program (B.1). Moreover, π_i^+ is always non-negative at the optimal solution. So relaxing π_i^+ to be a real number will not affect the optimal value. Thus, the linear program B.1 is equivilent to the following linear program:

$$\begin{array}{lll}
\max & u & \\
\text{s.t.} & u \le x_i U_+^d(i) + (1 - x_i - y_i - z_i)U_-^d(i) & \\
& \qquad +\pi_i^+ U_+^d(i) + z_i^- U_-^d(i) & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_+} p_e = x_i & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_{s+}} p_e = y_i & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}:e_i=\theta_{s-}} p_e = z_i & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}} p_{\mathbf{e}} = 1 & \\
& p_{\mathbf{e}} \ge 0 & \forall\, \mathbf{e} \in \mathcal{E} \\
& \pi_i^+ U_+^d(i) + z_i U_-^d(i) \le 0 & \forall\, i \in [n] \\
& \pi_i^+ \le y_i & \forall\, i \in [n]
\end{array} \tag{B.2}$$

The dual of LP (B.2) is the following LP.

$$\begin{array}{lll}
\min & \sum_{i=1}^n U_-^d(i) w_i + r & \\
\text{s.t.} & r \ge \sum_{i:e_i=\theta_+} \alpha_i + \sum_{i:e_i=\theta_{s+}} \beta_i + \sum_{i:e_i=\theta_{s-}} \gamma_i & \forall\, \mathbf{e} \in \mathcal{E} \\
& \alpha_i = [U_+^d(i) - U_-^d(i)]w_i & \forall\, i \in [n] \\
& \beta_i = \varphi_i - w_i U_-^d(i) & \forall\, i \in [n] \\
& \gamma_i = -\delta_i U_-^d(i) & \forall\, i \in [n] \\
& \sum_{\mathbf{e}\in\mathcal{E}} p_{\mathbf{e}} = 1 & \\
& \varphi_i = U_+^d(i) w_i - U_+^d(i)\delta_i & \forall\, i \in [n] \\
& \sum_{i=1}^n w_i = 1 &
\end{array} \tag{B.3}$$

in which $\alpha_i, \beta_i, \gamma_i$ correspond to the constraints defining x_i, y_i, z_i respectively. Note that

$$\alpha_i = [U_+^d(i) - U_-^d(i)]w_i \ge [U_+^d(i) - U_-^d(i)]w_i - U_+^d(i)\delta_i = \beta_i$$

Also, since $\varphi_i \ge 0$ and $U_+^d(i) \ge 0$ too, so we have the implicit constraint $w_i \ge \delta_i \ge 0$. Therefore,

$$\begin{aligned}
\beta_i &= [U_+^d(i) - U_-^d(i)]w_i - U_+^d(i)\delta_i \\
&\ge [U_+^d(i) - U_-^d(i)]\delta_i - U_+^d(i)\delta_i \\
&= -U_-^d(i)\delta_i = \gamma_i
\end{aligned}$$

Since $\gamma_i = -U_-^d(i)\delta_i \ge 0$, this implies

$$\alpha_i \ge \beta_i \ge \gamma_i \ge 0$$

Proof of Lemma 13

This is because when T is fixed, the weight of covering any target i by a sensor has been determined – either β_i if $i \in T^N$ or γ_i if $i \in T^c$. Therefore, to maximize the total weights, we simply pick the largest m elements in $\{\beta_i \mid i \in T^N\} \cup \{\gamma_i \mid i \in T^c\}$.

Proof of Theorem 6.2.2

The proof follows from the following two lemmas.

Lemma 29. *When $\alpha_i \geq \beta_i \geq \gamma_i \geq 0, \forall i \in [n]$, function $g(T)$ is nonnegative, monotone increasing and submodular.*

Proof of Lemma 29. It is easy to see that $g(T) \geq 0$ and is monotone increasing in T. We only prove its submodularity. Since $\sum_{i \in T} \alpha_i$ is a modular function of T, we only need to prove that function $f'(T) = \Sigma^m_{\max}\big(\{\beta_i \mid i \in T^N \cup T\} \cup \{\gamma_i \mid i \in T^c\}\big)$ is submodular in T. The key step is to prove that the following function is submodular:

$$W(S) = \textstyle\sum^m_{\max}(\{\beta_i | i \in S\} \cup \{\gamma_i | i \in \overline{S}\})$$

where $\beta_i \geq \gamma_i$ for all $i \in [n]$ and $\overline{S} = [n] - S$ is the complement of S. Notice that $W(T) \neq f'(T)$ (instead $W(T^N \cup T) = f'(T)$), so they are two different functions despite the similarity.

Pick any sets $S \subset T \subseteq [n]$ and $j \notin T$. Following the standard definition of submodularity, we prove the following inequality:

$$W(S \cup \{j\}) - W(S) \geq W(T \cup \{j\}) - W(T).$$

This follows a case analysis. For convenience, we will say "β_j [γ_j] contributes to $W(S)$" if β_j [γ_j] is among the largest m weights of $\{\beta_i | i \in S\} \cup \{\gamma_i | i \in \overline{S}\}$; Moreover, we denote set $S \cup \{j\}$ by S_{+j}.

- β_j contributes to $W(T_{+j})$. Then we must have that β_j also contributes to $W(S_{+j})$ since $S \subset T$. In this case, $W(S_{+j}) - W(S)$ equals β_j minus the smallest weight that contributes to $W(S)$. On the other hand, $W(T_{+j}) - W(T)$ equals β_j minus the smallest weight that contributes to $W(T)$. Since $S \subset T$, the smallest weight contributing to $W(T)$ is larger than the smallest weight contributing to $W(S)$. This implies $W(S_{+j}) - W(S) \geq W(T_{+j}) - W(T)$.

- β_j does not contribute to $W(T_{+j})$. In this case $W(T_{+j}) - W(T) = 0$ and $W(S_{+j}) - W(S) \geq 0$. Therefore, $W(S_{+j}) - W(S) \geq W(T_{+j}) - W(T)$.

As a result, $W(S)$ is submodular. We now show that $f'(T)$ is submodular by proving

$$f'(S_{+j}) - f'(S) \geq f'(T_{+j}) - f'(T)$$

for any $S \subset T \subseteq [n]$ and $j \notin T$. Let $A = T^N_{+j} \cup T_{+j} \setminus (T^N \cup T)$ and $B = S^N_{+j} \cup S_{+j} \setminus (S^N \cup S)$. Note that $A \subseteq B$ since $S \subset T$. Therefore

$$\begin{aligned} f'(S_{+j}) - f'(S) &= W(S^N_{+j} \cup S_{+j}) - W(S^N \cup S) \\ &= W(S^N \cup S \cup B) - W(S^N \cup S) \\ &\geq W(S^N \cup S \cup A) - W(S^N \cup S) \\ &\geq W(T^N \cup T \cup A) - W(T^N \cup T) \\ &= f'(T_{+j}) - f'(T), \end{aligned}$$

where the first inequality follows from monotonicity of function $W(S)$ and the second inequality follows from submodularity of $W(S)$. This proves that $f'(T)$, thus $f(T)$, is submodular. □

Lemma 30. *When $\alpha_i \geq \beta_i \geq \gamma_i \geq 0, \forall i \in [n]$, Algorithm 5 outputs a $\frac{1}{2}(1 - \frac{1}{e})$-approximation for the slave problem.*

Proof of Lemma 30. Let T^*_g and T^*_f be the optimal solution to maximizing $g(T)$ and $f(T)$ subject to $|T| \leq k$, respectively. Let $\widehat{T}$ be the set generated by the greedy process (step 2 – 5) in Algorithm 5. Our goal is to prove $f(\widehat{T}) \geq \frac{1}{2}(1 - \frac{1}{e}) f(T^*_f)$. The key step is to show the following relations:

$$f(T) \leq g(T) \leq 2f(T), \qquad \forall T \subseteq [n].$$

Since the $\Sigma^m_{\max}$ operator in $g(T)$ acts on a larger set than that in $f(T)$, this implies $g(T) \geq f(T)$. We now prove $g(T) \leq 2f(T)$. Since T, T^N, T^c are mutually disjoint, the weights that contribute to $f(T)$ are all indexed by different vertices. However, since $T \subseteq T^N \cup T$, there may exist vertex $i \in T$ such that both α_i and β_i contribute to $g(T)$. Let $A \subseteq T$ be all such i's. We have

$$\sum_{i \in A} \beta_i \leq \sum_{i \in A} \alpha_i \leq \sum_{i \in T} \alpha_i \leq f(T). \tag{B.4}$$

Moreover, if we remove the portion of $\sum_{i \in A} \beta_i$ from $g(T)$, then the left weights are all indexed by different vertices and their total weights are at most $f(T)$. That is,

$$g(T) - \sum_{i \in A} \beta_i \leq f(T) \tag{B.5}$$

Combining Inequalities (B.4) and (B.5) yields that $g(T) \leq f(T) + \sum_{i \in A} \beta_i \leq 2f(T)$, as desired.

By the monotone submodularity of $g(T)$ (Lemma 29), we have $g(\widehat{T}) \geq (1 - \frac{1}{e}) g(T^*_g)$. Since $g(T^*_g) \geq g(T^*_f) \geq f(T^*_f)$ and $2f(\widehat{T}) \geq g(\widehat{T})$, this implies $f(\widehat{T}) \geq \frac{1}{2}(1 - \frac{1}{e}) f(T^*_f)$.

□

B.2 Counter Example to Submodularity of $f(T)$

Recall that

$$f(T) = \sum_{i \in T} \alpha_i + \Sigma^m_{\max}\big(\{\beta_i \mid i \in T^N\} \cup \{\gamma_i \mid i \in T^c\}\big)$$

Consider a simple line graph G with 5 vertices, as in Figure B.1. Let $\tau = 1$ and $m = 2$. Moreover, $\alpha_i = \beta_i = 1$ while $\gamma_i = 0$ for all $i = 1, .., 5$.

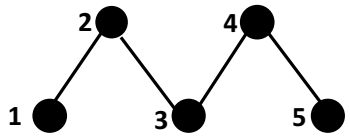


Figure B.1: Graph G for the counter example .

Consider $S = \{2\}$, $T = \{2, 4\}$ and $j = 1 \notin T$. We have $f(S) = 3, f(S \cup \{j\}) = 3, f(T) = 4$ and $f(T \cup \{j\}) = 5$. Therefore,

$$f(T \cup \{j\}) - f(T) = 1 > 0 = f(S \cup \{j\}) - f(S).$$

So $f(T)$ is not submodular in T.

Appendix C

Omitted Proofs From Section 6.3

C.1 Proof of Proposition 5

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 (6.10). 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 (6.10) and the equality is due to the first set of constraints in LP (6.10). 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.

C.2 Proof of Propositions in Section 6.3.2.2

Proof of Proposition 7

This is a slight modification from a proof of the hardness of Bayesian Stackelberg games (Theorem 2 in (Li et al., 2016)). 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.

Proof of Proposition 8

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: $p^\theta_{c,j}$ [$p^\theta_{u,j}$] 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 9). The following linear program, with variables $p^\theta_{c,j}$ and $\mathbf{x}$, computes the optimal defender utility.

$$
\begin{array}{lll}
\text{maximize} & \sum_{\theta\in\Theta} \lambda_\theta \sum_{s\in\{c,u\}} p^\theta_{s,i_\theta} U^d_x(i_\theta;\theta) & \\
\text{subject to} & \sum_{j\in\{0,i_\theta\}} p^\theta_{c,j} = x_{i_\theta}, & \text{for } \theta \in \Theta. \\
& \sum_{j\in\{0,i_\theta\}} p^\theta_{u,j} = 1 - x_{i_\theta}, & \text{for } \theta \in \Theta. \\
& \sum_{s\in\{c,u\}} p^\theta_{s,j} U^a_s(j;\theta) \geq \sum_{s\in\{c,u\}} p^\theta_{s,j} U^a_s(j';\theta), & \text{for } \theta \in \Theta. \\
& \mathbf{x} \in \mathcal{P} &
\end{array}
\tag{C.1}
$$

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 ($\mathcal{P}$ is the set of all implementable marginals.)

Proof of Proposition 9

LP Formulation of the Problem and Its Dual

Using similar notations as Section 6.3.3, 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

$$x = \mathbb{E}(e) = \sum_{e \in \mathcal{S}} e \times p_e \tag{C.2}$$

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

Since 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.[1]

$$\begin{array}{lll}
\text{maximize} & \sum_{\theta,i} \lambda_\theta \left[x^\theta_{ii} U^c_d(i;\theta) + (\alpha^\theta_i - x^\theta_{ii}) U^u_d(i;\theta) \right] & \\
\text{subject to} & x^\theta_{ii} U^c_a(i,\theta) + (\alpha^\theta_i - x^\theta_{ii}) U^u_a(i,\theta) \geq & \\
& \qquad x^\theta_{ij} U^c_a(j,\theta) + (\alpha^\theta_i - x^\theta_{ij}) U^u_a(j,\theta), & \text{for } i,j,\theta. \\
& \alpha^\theta_i = \sum_{e \in E} p^\theta_{e,i}, & \text{for } i,\theta. \\
& \sum_{e \in E} e \times p^\theta_{e,i} = x^\theta_i, & \text{for } i,\theta. \\
& \sum_{i=1}^n p^\theta_{e,i} = p_e, & \text{for } e,\theta. \\
& \sum_{s \in E} p_e = 1 & \\
& p^\theta_{e,i} \geq 0, p_e \geq 0, & \text{for } e,i,\theta.
\end{array} \tag{C.3}$$

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 (C.3). Instead of providing the exact dual program, we abstractly represent the dual by highlighting the non-trivial part, as follows:

$$\begin{array}{lll}
\text{minimize} & \gamma & \\
\text{subject to} & poly(n, |\Theta|) \text{ linear constraints on } y^\theta_i, \beta^\theta_i & \\
& -\beta^\theta_i + e \cdot y^\theta_i + q^\theta_e \geq 0, & \text{for } i,e,\theta. \\
& \sum_\theta -q^\theta_e + \gamma \geq 0, & \text{for } e.
\end{array} \tag{C.4}$$

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 (C.4). Notice that the first (implicitly described) constraint does not depend on γ, q^θ_e. So the last constraint, together

[1] 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.

with the "min" objective, yields that $\gamma = \max_{e \in E} \sum_\theta q_e^\theta$ at optimality. The middle constraint, together with the "min" objective, yields that $q_e^\theta = \max_i [\beta_i^\theta - e \cdot y_i^\theta]$ at optimality. As a result, the dual program can be re-written in the following form:

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

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

The Defender Oracle

The defender oracle problem is precisely to evaluate the function

$$f(y_i^\theta, \beta_i^\theta) = \max_{e \in E} \left[\sum_\theta \max_i (\beta_i^\theta - e \cdot y_i^\theta) \right] \tag{C.5}$$

for any given input $y_i^\theta, \beta_i^\theta$. When the attacker of type θ is only interested in a small number of targets, say a subset S of targets. Then in LP (C.3), the third constraint on $x_i^\theta \in \mathbb{R}^n$ 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_i^θ for all $i \notin S$; Moreover, for those $i \in S$, the constraint on x_i^θ 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_i^θ are non-zeros vectors only for those $i \in S$; moreover, the non-zero y_i^θ only has non-zeros at those entries corresponding to S.

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 $f(y_i^\theta, \beta_i^\theta)$ is NP-hard, even when only two y_i^θ's are non-zero vectors for each θ and each of these two y_i^θ'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_i^θ as follows: $y_{ii}^\theta = 1$, $y_{ij}^\theta = -1$ and $y_{ik}^\theta = 0$ for any $k \neq i, j$; define y_j^e as follows: $y_{ji}^\theta = -1$, $y_{jj}^\theta = 1$ and $y_{jk}^\theta = 0$ for any $k \neq i, j$. Let $\beta_i^\theta = 0$ 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

$$c(e) = \sum_{\theta \in \Theta} \max_k (\beta_k^\theta - e \cdot y_k^\theta) = \sum_{\theta = (i,j) \in \Theta} \max(-e \cdot y_i^\theta, -e \cdot y_j^\theta).$$

Note that $\max(-e \cdot y_i^\theta, -e \cdot y_j^\theta) = 1$ if and only if edge θ is cut by strategy e (in which case $e \cdot y_i^\theta, e \cdot y_j^\theta$ equals $1, -1$ respectively). Otherwise $\max(-e \cdot y_i^\theta, -e \cdot y_j^\theta) = 0$. Therefore, $c(e)$ equals precisely the cut size induced by e. Note that evaluating function f defined in Equation (C.5) 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.

C.3 Proof of the Polytope Transformation Lemma

In this section, we prove Lemma 15.

Part 1: This is standard, and can be found, e.g., in (Boyd & Vandenberghe, 2004). We provide a proof for completeness. Consider any two elements $(\mathbf{x}, p)$ and $(\mathbf{y}, q)$ from $\widetilde{\mathcal{P}}$. So there exists $\mathbf{a}, \mathbf{b} \in \mathcal{P}$ such that $\mathbf{x} = p \cdot \mathbf{a}$ and $\mathbf{y} = q \cdot \mathbf{b}$. To prove the convexity, we need to show $\alpha \cdot (\mathbf{x}, p) + \beta \cdot (\mathbf{y}, q) \in \widetilde{\mathcal{P}}$ for any $\alpha \in (0, 1)$ and $\alpha + \beta = 1$. If $p = q = 0$, this is obvious; Otherwise, we have

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

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

Part 2: First, it is easy to see that any element from $\widetilde{\mathcal{P}}$ 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 $p \geq 0$, $(\mathbf{x}, p) \in \widetilde{\mathcal{P}}$. It is easy to see that this is true for $p > 0$ since $x/p \in \mathcal{P}$. The non-trivial part is when $p = 0$, in which case $(\mathbf{x}, p) \in \widetilde{\mathcal{P}}$ 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 $\mathcal{P}$ is bounded. If (by contradiction) there exists $\mathbf{x}_0 \neq \mathbf{0}$ satisfying $A\mathbf{x}_0 \leq 0$, then for any $\mathbf{x} \in \mathcal{P}$, we must have $\mathbf{x} + \alpha\mathbf{x}_0 \in \mathcal{P}$ for any $\alpha > 0$, which contradicts the fact that $\mathcal{P}$ is bounded.

Part 3: If $\mathcal{P}$ has a separation oracle $\mathcal{O}$, then the following is a separation oracle for $\widetilde{\mathcal{P}}$. 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 $\mathbf{x}_0/p_0 \in \mathcal{P}$. 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 $\mathbf{x}' \in \mathcal{P}$. We claim that $\mathbf{a}^T \cdot \mathbf{x} - bp = 0$ is a hyperplane separating $(\mathbf{x}_0, p_0)$ from $\widetilde{\mathcal{P}}$. In particular, for any $(\mathbf{x}, p) \in \widetilde{\mathcal{P}}$ with $p > 0$, $\exists \mathbf{x}' \in \mathcal{P}$ such that $\mathbf{x}/p = \mathbf{x}'$. Note that $\mathbf{a}^T \cdot \mathbf{x}' \leq b$ since $\mathbf{x}' \in \mathcal{P}$, 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 $\mathcal{P}$ is bounded, we can find some $L_0 > 0$ large enough such that $\mathbf{y}_0 = L_0\mathbf{x}_0 \notin conv(\mathcal{P}, 0)$, where $conv(\mathcal{P}, 0)$ is the convex hull of $\mathcal{P}$ and the origin 0 (thus contains $\mathcal{P}$), and is introduced for technical convenience. Let $\mathbf{a}\cdot\mathbf{y} = b$ be a hyperplane separating $\mathbf{y}_0$ from $conv(\mathcal{P}, 0)$. That is $\mathbf{a}\cdot\mathbf{y}_0 > b$ and $\mathbf{a}\cdot\mathbf{y} \leq b$ for any $\mathbf{y} \in conv(\mathcal{P}, 0)$, in particular, for any $\mathbf{y} \in \mathcal{P}$. Similarly to the argument in case 2, we know that $\mathbf{a}\cdot\mathbf{x} \leq pb$ for any $(\mathbf{x}, p) \in \widetilde{\mathcal{P}}$. Note that, since $0 \in conv(\mathcal{P}, 0)$, 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 $\widetilde{\mathcal{P}}$.

Bibliography

Agmon, N., Sadov, V., Kaminka, G. A., & Kraus, S. (2008). The impact of adversarial knowledge on adversarial planning in perimeter patrol.. Vol. 1, pp. 55–62.

Agrawal, S., Ding, Y., Saberi, A., & Ye, Y. (2010). Correlation robust stochastic optimization. In *Proceedings of the Twenty-first Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '10, pp. 1087–1096, Philadelphia, PA, USA. Society for Industrial and Applied Mathematics.

Akerlof, G. A. (1970). The market for" lemons": Quality uncertainty and the market mechanism. *The Quarterly Journal of Economics*, *84*(3), 488–500.

Alaei, S., Fu, H., Haghpanah, N., Hartline, J. D., & Malekian, A. (2012). Bayesian optimal auctions via multi- to single-agent reduction.. In Faltings, B., Leyton-Brown, K., & Ipeirotis, P. (Eds.), *ACM Conference on Electronic Commerce*, p. 17. ACM.

Alon, N., Emek, Y., Feldman, M., & Tennenholtz, M. (2013). Adversarial leakage in games. *SIAM Journal on Discrete Mathematics*, *27*(1), 363–385.

Alonso, R., & Camara, O. (2014). Persuading voters. Working paper.

Alonso, R., & Câmara, O. (2016). Persuading voters. *American Economic Review*, *106*(11), 3590–3605.

Alpern, S., Morton, A., & Papadaki, K. (2011). Patrolling games. *Operations research*, *59*(5), 1246–1257.

An, B., Shieh, E., Tambe, M., Yang, R., Baldwin, C., DiRenzo, J., Maule, B., & Meyer, G. (2012). PROTECT–a deployed game theoretic system for strategic security allocation for the United States Coast Guard. *AI Magazine*, *33*(4), 96.

Anderson, S. P., & Renault, R. (2006). Advertising content. *American Economic Review*, *96*(1), 93–113.

Antioch, G. (2013). Persuasion is now 30 per cent of us gdp. *Economic Roundup*, pp. 1–10.

Arieli, I., & Babichenko, Y. (2016). Private Bayesian persuasion. *Available at SSRN 2721307*.

Aumann, R., Maschler, M., & Stearns, R. (1995). *Repeated Games with Incomplete Information*. MIT Press.

Babaioff, M., Kleinberg, R., & Paes Leme, R. (2012). Optimal mechanisms for selling information. In *Proceedings of the 13th ACM Conference on Electronic Commerce*, EC '12, pp. 92–109, New York, NY, USA. ACM.

Babichenko, Y., & Barman, S. (2017). Algorithmic aspects of private Bayesian persuasion. In *Proceedings of the 2017 ACM Conference on Innovations in Theoretical Computer Science*, ITCS.

Bardhi, A., & Guo, Y. (2016). Modes of persuasion toward unanimous consent. *Working paper*.

Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W., & Vance, P. H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations research*, *46*(3), 316–329.

Basilico, N., Celli, A., De Nittis, G., & Gatti, N. (2017a). Coordinating multiple defensive resources in patrolling games with alarm systems. In *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*, AAMAS.

Basilico, N., De Nittis, G., & Gatti, N. (2017b). Adversarial patrolling with spatially uncertain alarm signals. *Artificial Intelligence*, *246*, 220–257.

Basilico, N., Gatti, N., & Amigoni, F. (2009a). Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '09.

Basilico, N., Gatti, N., Rossi, T., Ceppi, S., & Amigoni, F. (2009b). Extending algorithms for mobile robot patrolling in the presence of adversaries to more realistic settings. In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology - Volume 02*, WI-IAT '09. IEEE Computer Society.

Bergemann, D., & Bonatti, A. (2015). Selling cookies. *American Economic Journal: Microeconomics*, *7*(3), 259–94.

Bergemann, D., Bonatti, A., & Smolin, A. (2016). The designing and pricing information. *Working Paper*.

Bergemann, D., Brooks, B., & Morris, S. (2015). The limits of price discrimination. *American Economic Review*, *105*(3), 921–57.

Bergemann, D., & Morris, S. (2016). Bayes correlated equilibrium and the comparison of information structures in games. *Theoretical Economics*, *11*(2), 487–522.

Bhaskar, U., Cheng, Y., Ko, Y. K., & Swamy, C. (2016). Hardness results for signaling in Bayesian zero-sum and network routing games. In *Proceedings of the 2016 ACM Conference on Economics and Computation (EC)*. ACM.

Border, K. (2007). Reduced Form Auctions Revisited. *Economic Theory*, *31*(1), 167–181.

Border, K. C. (1991). Implementation of Reduced Form Auctions: A Geometric Approach. *Econometrica*, *59*(4).

Bosansky, B., Jiang, A. X., Tambe, M., & Kiekintveld, C. (2015). Combining compact representation and incremental generation in large games with sequential strategies. In *AAAI*.

Bošanský, B., Kiekintveld, C., Lisý, V., & Pěchouček, M. (2014). An exact double-oracle algorithm for zero-sum extensive-form games with imperfect information. *J. Artif. Int. Res.*, *51*(1), 829–866.

Bošanský, B., Lisý, V., Jakob, M., & Pěchouček, M. (2011). Computing time-dependent policies for patrolling games with mobile targets.. In *AAMAS*. IFAAMAS.

Boyd, S., & Vandenberghe, L. (2004). *Convex Optimization*. Cambridge University Press, New York, NY, USA.

Brocas, I., & Carrillo, J. D. (2007). Influence through ignorance. *The RAND Journal of Economics*, *38*(4), 931–947.

Brown, G., Carlyle, M., Diehl, D., Kline, J., & Wood, K. (2005). A two-sided optimization for theater ballistic missile defense. *Oper. Res.*, *53*(5), 745–763.

Brown, M., Sinha, A., Schlenker, A., & Tambe, M. (2016). One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In *AAAI conference on Artificial Intelligence (AAAI)*.

Cai, Y., Daskalakis, C., & Weinberg, S. M. (2012). An algorithmic characterization of multi-dimensional mechanisms. In *Proceedings of the Forty-fourth Annual ACM Symposium on Theory of Computing*, STOC '12, pp. 459–478, New York, NY, USA. ACM.

Calinescu, G., Chekuri, C., Pál, M., & Vondrák, J. (2011). Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*.

Carthy, S. M., Tambe, M., Kiekintveld, C., Gore, M. L., & Killion, A. (2016). Preventing illegal logging: simultaneous optimization of resource teams and tactics for security. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, pp. 3880–3886. AAAI Press.

Cermak, J., Bosansky, B., Durkota, K., Lisy, V., & Kiekintveld, C. (2016). Using correlated strategies for computing stackelberg equilibria in extensive-form games. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Cermak, J., Bošanský, B., & Lisý, V. (2017). An algorithm for constructing and solving imperfect recall abstractions of large extensive-form games. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 936–942.

Chakraborty, A., & Harbaugh, R. (2014). Persuasive puffery. *Marketing Science*, *33*(3), 382–400.

Chan, J., Gupta, S., Li, F., & Wang, Y. (2016). Pivotal persuasion. *Available at SSRN*.

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

Cheng, Y., Cheung, H. Y., Dughmi, S., Emamjomeh-Zadeh, E., Han, L., & Teng, S.-H. (2015). Mixture selection, mechanism design, and signaling. In *Foundations of Computer Science (FOCS), 2015 IEEE 56th Annual Symposium on*, pp. 1426–1445. IEEE.

Colbourn, J. C., Provan, J. S., & Vertigan, D. (1995). The complexity of computing the tutte polynomial on transversal matroids. *Combinatorica*.

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

Conitzer, V., & Sandholm, T. (2006). Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM conference on Electronic commerce*, pp. 82–90. ACM.

Crawford, P., & Sobel, J. (1982). Strategic information transmission. *Econometrica*.

Crawford, V. (1998). A survey of experiments on communication via cheap talk. *Journal of Economic theory*, *78*(2), 286–298.

Cryan, M., & Dyer, M. (2002). A polynomial-time algorithm to approximately count contingency tables when the number of rows is constant. In *Proceedings of the Thiry-fourth Annual ACM Symposium on Theory of Computing*, STOC '02, pp. 240–249, New York, NY, USA. ACM.

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

Dughmi, S. (2014). On the hardness of signaling. In *Proceedings of the 55th Symposium on Foundations of Computer Science*, FOCS '14. IEEE Computer Society.

Dughmi, S. (2017). Algorithmic information structure design: a survey. *ACM SIGecom Exchanges*, *15*(2), 2–24.

Dughmi, S., Immorlica, N., & Roth, A. (2014). Constrained signaling in auction design. In *Proceedings of the Twenty-five Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '14. Society for Industrial and Applied Mathematics.

Dyer, M. (2003). Approximate counting by dynamic programming. In *Proceedings of the Thirty-fifth Annual ACM Symposium on Theory of Computing*, STOC '03, pp. 693–699, New York, NY, USA. ACM.

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

Fang, F., Jiang, A. X., & Tambe, M. (2013). Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Fang, F., Nguyen, T. H., Pickles, R., Lam, W. Y., Clements, G. R., An, B., Singh, A., & Tambe, M. (2016a). Deploying paws to combat poaching: Game-theoretic patrolling in areas with complex terrain (demonstration). In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pp. 4355–4356. AAAI Press.

Fang, F., Nguyen, T. H., Pickles, R., Lam, W. Y., Clements, G. R., An, B., Singh, A., Tambe, M., & Lemieux, A. (2016b). Deploying PAWS: Field optimization of the protection assistant for wildlife security. In *IAAI*.

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

Garey, M. R., & Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness (Series of Books in the Mathematical Sciences)* (First Edition edition). W. H. Freeman.

Gentzkow, M., & Kamenica, E. (2014). Costly persuasion. *American Economic Review*, *104*(5), 457–62.

Gentzkow, M., & Kamenica, E. (2016). Competition in persuasion. *The Review of Economic Studies*, *84*(1), 300–322.

Gholami, S., Ford, B., Fang, F., Plumptre, A., Tambe, M., Driciru, M., Wanyama, F., Rwetsiba, A., Nsubaga, M., & Mabonga, J. (2017). Taking it for a test drive: a hybrid spatio-temporal model for wildlife poaching prediction evaluated through a controlled field test. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 292–304. Springer.

Gick, W., & Pausch, T. (2012). *Persuasion by stress testing: Optimal disclosure of supervisory information in the banking sector*. No. 32/2012. Discussion Paper, Deutsche Bundesbank.

Goldstein, I., & Leitner, Y. (2013). Stress tests and information disclosure..

Gopalan, P., Nisan, N., & Roughgarden, T. (2015). Public projects, boolean functions, and the borders of border's theorem. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, EC '15, pp. 395–395, New York, NY, USA. ACM.

Grötschel, M., Lovász, L., & Schrijver, A. (1988). *Geometric Algorithms and Combinatorial Optimization*, Vol. 2 of *Algorithms and Combinatorics*. Springer.

Guo, M., & Deligkas, A. (2013). Revenue maximization via hiding item attributes. *CoRR*, *abs/1302.5332*.

Guo, Q., An, B., Bosansky, B., & Kiekintveld, C. (2017). Comparing strategic secrecy and stackelberg commitment in security games. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*.

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

Jain, M., Korzhyk, D., Vaněk, O., Conitzer, V., Pěchouček, M., & Tambe, M. (2011). A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems - Volume 1*, AAMAS '11, pp. 327–334.

Jain, M., Leyton-Brown, K., & Tambe, M. (2012). The deployment-to-saturation ratio in security games. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, pp. 1362–1370. AAAI Press.

Jerrum, M. R., Valiant, L. G., & Vazirani, V. V. (1986). Random generation of combinatorial structures from a uniform distribution. *Theor. Comput. Sci.*, *43*, 169–188.

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

Johnson, J. P., & Myatt, D. P. (2006). On the simple economics of advertising, marketing, and product design. *American Economic Review*, *96*(3), 756–784.

Kahn, J., & Kayll, P. M. (1997). On the stochastic independence properties of hard-core distributions. *Combinatorica*, *17*(3), 369–391.

Kamenica, E., & Gentzkow, M. (2011). Bayesian persuasion. *American Economic Review*, *101*(6), 2590–2615.

Khot, S., & Saket, R. (2012). Hardness of finding independent sets in almost q-colorable graphs. In *IEEE 53rd Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 380–389. IEEE.

Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Ordóñez, F., & Tambe, M. (2009). Computing optimal randomized resource allocations for massive security games. In *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems-Volume 1*, pp. 689–696. International Foundation for Autonomous Agents and Multiagent Systems.

Knuth, D. E. (1997). *The art of computer programming*, Vol. 3. Pearson Education.

Kolotilin, A. (2015). Experimental design to persuade. *Games and Economic Behavior*, *90*, 215–226.

Kolotilin, A., Mylovanov, T., Zapechelnyuk, A., & Li, M. (2017). Persuasion of a privately informed receiver. *Econometrica*, *85*(6), 1949–1964.

Korzhyk, D., Conitzer, V., & Parr, R. (2011a). Security games with multiple attacker resources. In *Twenty-Second International Joint Conference on Artificial Intelligence*.

Korzhyk, D., Yin, Z., Kiekintveld, C., Conitzer, V., & Tambe, M. (2011b). Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *Journal of Artificial Intelligence Research*, *41*, 297–327.

Kremer, I., Mansour, Y., & Perry, M. (2014). Implementing the "wisdom of the crowd". *Journal of Political Economy*, *122*(5), 988–1012.

Letchford, J., & Conitzer, V. (2010). Computing optimal strategies to commit to in extensive-form games. In *Proceedings of the 11th ACM conference on Electronic commerce*, pp. 83–92. ACM.

Letchford, J., Conitzer, V., & Munagala, K. (2009). Learning and approximating the optimal strategy to commit to.. In Mavronicolas, M., & Papadopoulou, V. G. (Eds.), *SAGT*, Vol. 5814 of *Lecture Notes in Computer Science*, pp. 250–262. Springer.

Li, Y., Conitzer, V., & Korzhyk, D. (2016). Catcher-evader games. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, pp. 329–337. AAAI Press.

Mansour, Y., Slivkins, A., & Syrgkanis, V. (2015). Bayesian incentive-compatible bandit exploration. *arXiv preprint arXiv:1502.04147*.

Mersheeva, V., & Friedrich, G. (2015). Multi-uav monitoring with priorities and limited energy resources. In *Proceedings of the Twenty-Fifth International Conference on Automated Planning and Scheduling*, pp. 347–355. AAAI Press.

Miltersen, P. B., & Sheffet, O. (2012). Send mixed signals: earn more, work less.. In Faltings, B., Leyton-Brown, K., & Ipeirotis, P. (Eds.), *ACM Conference on Electronic Commerce*, pp. 234–247. ACM.

Moreto, W. (2013). *To conserve and protect: Examining law enforcement ranger culture and operations in Queen Elizabeth National Park, Uganda*. Ph.D. thesis, Rutgers University-Graduate School-Newark.

Nguyen, T. H., Delle Fave, F. M., Kar, D., Lakshminarayanan, A. S., Yadav, A., Tambe, M., Agmon, N., Plumptre, A. J., Driciru, M., Wanyama, F., et al. (2015). Making the most

of our regrets: Regret-based solutions to handle payoff uncertainty and elicitation in green security games. In *International Conference on Decision and Game Theory for Security*, pp. 170–191. Springer.

Nudelman, E., Wortman, J., Shoham, Y., & Kevin, L.-B. (2004). 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*, pp. 880–887. IEEE Computer Society.

Nyirenda, V. R., & Chomba, C. (2012). Field foot patrol effectiveness in kafue national park, zambia. *Journal of Ecology and the Natural Environment*, *4*(6), 163–172.

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

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

Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., & Kraus, S. (2008a). Deployed ARMOR protection: the application of a game theoretic model for security at the los angeles international airport. In *AAMAS: industrial track*.

Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., & Kraus, S. (2008b). 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*, pp. 125–132. International Foundation for Autonomous Agents and Multiagent Systems.

Pitowsky, I. (1991). Correlation polytopes: Their geometry and complexity.. *Math. Program.*, 395–414.

Powell, R. (2007). Allocating defensive resources with private information about vulnerability. *American Political Science Review*, *101*(04), 799–809.

Rabinovich, Z., Jiang, A. X., Jain, M., & Xu, H. (2015). Information disclosure as a means to security. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems, AAMAS, Istanbul, Turkey, 2015*.

Rayo, L., & Segal, I. (2010). Optimal information disclosure. *Journal of Political Economy*, *118*(5), 949 – 987.

Rowe, N. C. (2006). A taxonomy of deception in cyberspace..

Rowe, N. C., & Rothstein, H. (2004). Two taxonomies of deception for attacks on information systems. *Journal of Information Warfare*, *3*(2), 27–39.

Rubinstein, A. (2017). Honest signaling in zero-sum games is hard...and lying is even harder!. In *Proceedings of the 44th international colloquium conference on Automata, Languages, and Programming*. Springer-Verlag.

Schrijver, A. (2003). *Combinatorial Optimization - Polyhedra and Efficiency*. Springer.

Singh, M., & Vishnoi, N. K. (2013). Entropy, optimization and counting. *CoRR*.

Stranders, R., De Cote, E. M., Rogers, A., & Jennings, N. R. (2013). Near-optimal continuous patrolling with teams of mobile information gathering agents. *Artificial intelligence*, *195*, 63–105.

Talmor, N., & Agmon, N. (2017). On the power and limitations of deception in multi-robot adversarial patrolling. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pp. 430–436.

Tambe, M. (2011). *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press.

Taneva, I. A. (2015). Information design..

Tsai, J., Rathi, S., Kiekintveld, C., Ordonez, F., & Tambe, M. (2009). Iris - a tool for strategic security allocation in transportation networks. In *The Eighth International Conference on Autonomous Agents and Multiagent Systems - Industry Track*.

Tsai, J., Yin, Z., young Kwak, J., Kempe, D., Kiekintveld, C., & Tambe, M. (2010). Urban security: Game-theoretic resource allocation in networked physical domains. In *National Conference on Artificial Intelligence (AAAI)*.

Valiant, L. G. (1979). The complexity of computing the permanent. *Theoretical computer science*, *8*(2), 189–201.

von Stackelberg, H. (1934). *Marktform und Gleichgewicht*. Springer, Vienna.

von Stengel, B., & Zamir, S. (2004). Leadership with commitment to mixed strategies.. *CDAM Research Report LSE-CDAM-2004-01, London School of Economics*.

Vorobeychik, Y., An, B., & Tambe, M. (2012). Adversarial patrolling games. In *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems-Volume 3*, pp. 1307–1308. International Foundation for Autonomous Agents and Multiagent Systems.

Waldfogel, J., & Chen, L. (2006). Does information undermine brand? information intermediary use and preference for branded web retailers. *The Journal of Industrial Economics*, *54*(4), 425–449.

Wang, Y. (2015). Bayesian persuasion with multiple receivers. *Available at SSRN 2625399*.

Weinberg, S. M. (2014). *Algorithms for Strategic Agents*. Ph.D. thesis, Massachusetts Institute of Technology.

Wittemyer, G., Northrup, J. M., Blanc, J., Douglas-Hamilton, I., Omondi, P., & Burnham, K. P. (2014). Illegal killing for ivory drives global decline in african elephants. *Proceedings of the National Academy of Sciences*, *111*(36), 13117–13121.

Xu, H. (2016). The mysteries of security games: Equilibrium computation becomes combinatorial algorithm design. In *Proceedings of the 2016 ACM Conference on Economics and Computation*, pp. 497–514. ACM.

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

Xu, H., Jiang, A. X., Sinha, A., Rabinovich, Z., Dughmi, S., & Tambe, M. (2015). 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*, pp. 674–680.

Yin, Y., An, B., Vorobeychik, Y., & Zhuang, J. (2013). Optimal deceptive strategies in security games: A preliminary study..

Yin, Y., Xu, H., Gain, J., An, B., & Jiang, A. X. (2015). Computing optimal mixed strategies for security games with dynamic payoffs. In *IJCAI*.

Yin, Z., Jiang, A. X., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., & Sullivan, J. P. (2012). TRUSTS: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, *33*(4), 59.

Zhuang, J., & Bier, V. M. (2010). Reasons for secrecy and deception in Homeland-Security resource allocation. *Risk Analysis*, *30*(12), 1737–1743.

Zhuang, J., & Bier, V. M. (2011). Secrecy and deception at equilibrium, with applications to anti-terrorism resource allocation. *Defence and Peace Economics*, *22*(1), 43–61.