

The Mysteries of Security Games: Equilibrium Computation Becomes Combinatorial Algorithm Design

HAIFENG XU, University of Southern California, Los Angeles CA

The *security game* is a basic model for resource allocation in adversarial environments. Here there are two players, a *defender* and an *attacker*. The defender wants to allocate her limited resources to defend critical targets and the attacker seeks his most favorable target to attack. In the past decade, there has been a surge of research interest in analyzing and solving security games that are motivated by applications from various domains. Remarkably, these models and their game-theoretic solutions have led to real-world deployments in use by major security agencies like the LAX airport, the US Coast Guard and Federal Air Marshal Service, as well as non-governmental organizations. Among all these research and applications, equilibrium computation serves as a foundation.

This paper examines security games from a theoretical perspective and provides a unified view of various security game models. In particular, each security game can be characterized by a set system \mathcal{E} which consists of the defender's pure strategies; The defender's best response problem can be viewed as a combinatorial optimization problem over \mathcal{E} . Our framework captures most of the basic security game models in the literature, including all the deployed systems; The set system \mathcal{E} arising from various domains encodes standard combinatorial problems like bipartite matching, maximum coverage, min-cost flow, packing problems, etc. Our main result shows that equilibrium computation in security games is essentially a combinatorial problem. In particular, we prove that, for any set system \mathcal{E} , the following problems can be reduced to *each other* in polynomial time: (0) combinatorial optimization over \mathcal{E} ; (1) computing the minimax equilibrium for zero-sum security games over \mathcal{E} ; (2) computing the strong Stackelberg equilibrium for security games over \mathcal{E} ; (3) computing the best or worst (for the defender) Nash equilibrium for security games over \mathcal{E} . Therefore, the hardness [polynomial solvability] of any of these problems implies the hardness [polynomial solvability] of all the others. Here, by "games over \mathcal{E} " we mean the class of security games with arbitrary payoff structures, but a fixed set \mathcal{E} of defender pure strategies. This shows that the complexity of a security game is essentially determined by the set system \mathcal{E} . We view drawing these connections as an important conceptual contribution of this paper.

Additional Key Words and Phrases: equilibrium computation; computational complexity; security games.

1. INTRODUCTION

The security of critical infrastructures and areas 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 attempt to address this fundamental question, which has led to the development of the well-known model of security games. A *security game* is a two-player game played between a *defender* and an *attacker*. The defender allocates (possibly randomly) limited security resources, subject to various domain constraints, to protect a set of *targets*; The attacker chooses one

Author's email: haifengx@usc.edu. A full version of this paper is available at <http://arxiv.org/abs/1603.02377>.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

EC'16, July 24–28, 2016, Maastricht, The Netherlands. Copyright © 2016 ACM 978-1-4503-3936-0/16/07 ...\$15.00.

<http://dx.doi.org/10.1145/2940716.2940796>

target to attack. This 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 passenger screening [Brown et al. 2016] and by non-governmental organizations in Malaysia for wildlife protection [Fang et al. 2016].

Equilibrium computation is perhaps the most basic problem in security games. Indeed, there have been numerous algorithms developed for solving various security games motivated by different real-world applications (we refer the reader to [Tambe 2011] for a review). However, many of these algorithms are based on integer linear programs and heuristics, which may run in exponential time or output non-optimal solutions. The computational complexity of solving these games is not well-understood. Moreover, most of the literature has focused on the computation of the strong Stackelberg equilibrium (minimax equilibrium when the game is zero-sum), which may be inappropriate when the players move simultaneously (see Section 3.2 for a more detailed discussion). In this paper, we systematically study the computational complexity of the three main equilibrium concepts adopted in security games, namely, the minimax equilibrium, strong Stackelberg equilibrium, and Nash equilibrium. However, instead of examining all the models one by one, we provide a unified view of security games that captures most of the basic models in the literature, and prove our results in this general framework. Interestingly, it turns out that none of these equilibrium concepts is computationally harder than the others in any security game captured by our framework.

1.1. Our Results

We start with a unified formulation of security games. In particular, we show that security games are essentially *bilinear* games, in which each player’s payoff has the form $\mathbf{x}^T \mathbf{A} \mathbf{y} + \alpha \cdot \mathbf{y}$; the defender’s mixed strategy \mathbf{x} lies in a polytope $\mathcal{P} \subseteq \mathbb{R}^n$ and the attacker’s mixed strategy \mathbf{y} is in the n -dimensional simplex Δ_n . Interestingly, the vertices of \mathcal{P} , i.e., all the defender pure strategies, form a set system \mathcal{E} , and the defender’s best response problem can be viewed as a combinatorial optimization problem over \mathcal{E} . This general framework captures most of the basic security game models in the literature, including all the deployed security systems. We show that, the set system \mathcal{E} arising from various security domains encodes many standard combinatorial problems like bipartite matching, maximum coverage, min-cost flow, packing problems, etc.

We are interested in solving the class of security games *over* \mathcal{E} , by which we mean all security games with arbitrary payoff structures, but a fixed set \mathcal{E} of defender pure strategies. Our main theoretical results build connections between combinatorial optimization over \mathcal{E} and equilibrium computation for security games over \mathcal{E} . In particular, we prove that, for any set system \mathcal{E} , the following problems can be reduced to *each other* in polynomial time: (0) combinatorial optimization over \mathcal{E} ; (1) computing the minimax equilibrium for zero-sum security games over \mathcal{E} ; (2) computing the strong Stackelberg equilibrium for security games over \mathcal{E} ; (3) computing the best or worst (for the defender) Nash equilibrium for security games over \mathcal{E} . Therefore, the hardness [polynomial solvability] of any of these problems implies the hardness [polynomial solvability] of all the others. This shows that the complexity of a security game is essentially determined by the set system \mathcal{E} . As applications of these results, we also show how to use them to easily recover and strengthen some known complexity results in the literature, as well as to resolve some open problems from previous work. We remark that

though these results are primarily theoretical, our reductions from equilibrium computation to combinatorial optimization can be practically implemented via well-known engineering approaches, e.g., the column generation technique.

1.2. Related Work

Several papers have examined the computational complexity of security games in particular settings. The most relevant are the following two papers: [Korzhyk et al. 2010] consider the security settings where each security resource can be allocated to protect a subset of targets; [Letchford and Conitzer 2013] consider security games on graphs where targets are nodes and security resources patrol along paths. They show polynomial solvability or NP-hardness under different conditions. To the best of our knowledge, there is no other work which specifically focuses on a complexity study of security games. Nevertheless, some hardness results are provided separately in different work for different models, e.g., [Brown et al. 2016; Gan et al. 2015]. We note that our framework only concerns the basic security game models. There are various refinements of the basic models, e.g., the Bayesian setting, repeated setting, stochastic setting, etc. Examining the complexity of these settings is an interesting avenue for future work, but is not the focus of this paper.

Also related to our work is the rich literature on equilibrium computation for succinctly represented games. The most fundamental problem along this line is to compute one Nash equilibrium for a two-player normal-form game. This is proven to be PPAD-hard [Chen et al. 2009; Daskalakis et al. 2009]. In the same setting, computing the Nash equilibrium that maximizes one player’s utility is NP-hard [Conitzer and Sandholm 2008; Gilboa and Zemel 1989], but the strong Stackelberg equilibrium can be computed in polynomial time by solving linear programs [Conitzer and Sandholm 2006]. [Immorlica et al. 2011] consider the computation of bilinear zero-sum games, and show how to compute the minimax equilibrium when both players’ action polytopes have explicit polynomial-size representations. They also reduce computing an ϵ -minimax equilibrium to an additive FPTAS for the player’s best response problem, using the no regret learning framework. However, they do not consider the reverse direction, namely, the reduction from best response to equilibrium computation. [Garg et al. 2011] consider bilinear general-sum games, and show that such games are general enough to capture many interesting classes of games, hence are hard to solve in general. They propose polynomial time algorithms for computing approximate equilibria when the payoff matrices have low rank.

2. PRELIMINARIES

Bilinear Game Basics. A bilinear game is given by a pair of matrices (A, B) and polytopes $(\mathcal{P}, \mathcal{Q})$. Given that player 1 plays $x \in \mathcal{P}$ and player 2 plays $y \in \mathcal{Q}$, the utilities for player 1 and 2 are $x^T A y$ and $x^T B y$ respectively. As we will show later, security games are essentially bilinear games with slightly richer structure of the form $x^T A y + \alpha \cdot y$ and $x^T B y + \beta \cdot y$. Note that each vertex of the polytope is a pure strategy, and a player may have exponentially many pure strategies even though her action polytope (e.g., a hypercube) can be compactly represented. When $A = -B$, the game is *zero-sum*. The definition of minimax equilibrium, Nash equilibrium, and strong Stackelberg equilibrium (SSE) for two-player normal-form games naturally generalize here.

Linear Optimization and Convex Decomposition via Oracles. Any linear optimization problem can be implicitly described as $\max_{x \in \mathcal{P}} c^T x$ where $\mathcal{P} \subseteq \mathbb{R}^N$ is a polytope. By “(linear) optimization over \mathcal{P} ” we mean solving the problem $\max_{x \in \mathcal{P}} c^T x$ for any $c \in \mathbb{R}^N$. A *membership oracle* for \mathcal{P} is an algorithm that, on input $x_0 \in \mathbb{R}^N$, correctly asserts whether x_0 is in \mathcal{P} or not. A *separation oracle* for \mathcal{P} is an algorithm that,

on input $x_0 \in \mathbb{R}^N$, either asserts $x_0 \in \mathcal{P}$ or finds a hyperplane $a^T x = b$ that separates x_0 from \mathcal{P} in the following sense: $a^T x_0 > b$ and $a^T x \leq b$ for any $x \in \mathcal{P}$. The membership [separation] problem for polytope \mathcal{P} is to compute a membership [separation] oracle for \mathcal{P} . In the following theorems, by “polynomial running time” we mean polynomial in the dimension N and the description length of the instance; Solving an LP means returning the optimal objective value as well as a vertex optimal solution. The following celebrated results are due to Grötschel, Lovász and Schrijver [Grötschel et al. 2012].

THEOREM 2.1. *Let $\mathcal{P} \subseteq \mathbb{R}^N$ be a polytope. If there is a polynomial time algorithm to solve the separation problem for \mathcal{P} , then there is a polynomial time algorithm to solve any linear program over \mathcal{P} as well as its dual program.*

THEOREM 2.2. *Let $\mathcal{P} \subseteq \mathbb{R}^N$ be a polytope, and let $x_0 \in \mathcal{P}$ be a point known to any algorithm. There is a polynomial time algorithm for the optimization problem over \mathcal{P} if and only if there is a polynomial time algorithm for the membership problem for \mathcal{P} .*

To implement a mixed strategy $x \in \mathcal{P}$, we need to decompose x into a linear combination of pure strategies, i.e., vertices of \mathcal{P} . This can also be done efficiently given access to an efficient separation oracle.

THEOREM 2.3. *Let $\mathcal{P} \subseteq \mathbb{R}^N$ be a polytope. If the separation problem for \mathcal{P} can be solved in polynomial time, then there is a polynomial time algorithm that, on any input $x \in \mathcal{P}$, computes $N + 1$ vertices $x_1, \dots, x_{N+1} \in \mathcal{P}$ and convex coefficients $\lambda_1, \dots, \lambda_{N+1}$ such that $x = \sum_{i=1}^{N+1} \lambda_i x_i$.*

3. THE MODEL OF SECURITY GAMES

3.1. Strategies and Payoff Structures

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 3.3.

A more convenient representation of a pure strategy, as will be used throughout the paper, is a *binary* vector $e \in \{0, 1\}^n$, 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 p over the elements in \mathcal{E} . The attacker chooses one target to attack, thus an *attacker pure strategy* is a target $i \in [n]$. We use $y \in \Delta_n$ to denote an attacker mixed strategy where y_i is the probability of attacking target i .

The payoff structure of the game is as follows: given that the attacker attacks target i , the defender gets a reward r_i if target i is covered or a cost c_i if i is uncovered; while the attacker gets a cost ζ_i if target i is covered or a reward ρ_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: $r_i > c_i$ and $\rho_i > \zeta_i$ for all $i \in [n]$. That is, covering a target is strictly beneficial to the defender than uncovering

it; and the attacker prefers to attack a target when it is uncovered.¹ We formalize the model of security games in the following definition.

Definition 3.1. [Security Game] A security game \mathcal{G} with n targets is given by the following tuple $(\mathbf{r}, \mathbf{c}, \rho, \zeta, \mathcal{E})$ and satisfies $r_i > c_i$ and $\rho_i > \zeta_i$ for all $i \in [n]$. The security game is *zero-sum* if $r_i + \zeta_i = 0$ and $c_i + \rho_i = 0$ for all $i \in [n]$.

We denote a security game by $\mathcal{G}(\mathbf{r}, \mathbf{c}, \rho, \zeta, \mathcal{E})$. When the game is zero-sum, we also use $\mathcal{G}(\mathbf{r}, \mathbf{c}, \mathcal{E})$ for short. We are interested in solving security games *over* \mathcal{E} , by which we mean security games with arbitrary payoff structures, but a fixed set \mathcal{E} of defender pure strategies. The defender's utility, as a function of the defender pure strategy \mathbf{e} and attacker pure strategy i , can be formally expressed as

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

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

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

where

$$x_i = \sum_{\mathbf{e} \in \mathcal{E}} p_e e_i \in [0, 1] \quad (2)$$

is the *marginal coverage probability* of target i . Let $\mathbf{x} = (x_1, \dots, x_n)^T$ 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.² Equation (1) shows that the defender's expected utility can be compactly expressed as the bilinear form $\mathbf{x}^T A \mathbf{y} + \alpha \cdot \mathbf{y}$ for some *non-negative* diagonal matrix A , where \mathbf{x} is the marginal probability induced by the defender mixed strategy. We note that 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.³

For the rest of this paper, we will simply 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 $U^a(\mathbf{x}, \mathbf{y})$ also has the bilinear form $\mathbf{x}^T B \mathbf{y} + \beta \cdot \mathbf{y}$ for some *non-positive* diagonal matrix B .

$$U^a(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n y_i \left(\rho_i \cdot [1 - x_i] + \zeta_i \cdot x_i \right). \quad (3)$$

¹In 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 the case here.

²Here we assume security forces have perfect protection effectiveness. That is, once a target is covered, regardless by one or multiple resources, it is fully protected with probability 1. Generalization to nonperfect effectiveness is straightforward.

³One type of security games that is not captured by this model is the network interdiction game [Tsai et al. 2010; Washburn and Wood 1995], in which the defender chooses edges to defend and the attacker chooses a path to attack. The task of the defender is to interdict the attacker at a certain edge. The utility functions in this game do not have the bilinear structure, and are non-convex in general. This belongs to the more general class of succinctly represented games with non-linear payoffs, which is EXP-hard to solve even in zero-sum cases [Feigenbaum et al. 1995].

3.2. Equilibrium Concepts

Many security games, including some deployed systems [An et al. 2012; Yin et al. 2012], are modeled as zero-sum games. That is, the defender’s reward [cost] is the negative of 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 is naturally zero-sum: 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 *minimax equilibrium*, and our goal is to compute the minimax equilibrium in $\text{poly}(n)$ time.

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 and 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, 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 trivial). Notice that, the attacker is not able to observe the defender’s real-time deployment (i.e., the sampled pure strategy) since he has to plan the attack before the defender’s real-time pure strategy is sampled.

The strong Stackelberg equilibrium (SSE) is appropriate only when the attacker does surveillance and can indeed observe the defender’s past actions. However, in many cases the attacker does little surveillance. In fact, even if the attacker intends to do surveillance, sometimes he cannot observe the defender’s strategies due to limited resources and confidentiality of the defender’s resource allocation (e.g., plainclothes police). In these settings, the defender cannot commit to a strategy, thus *Nash equilibrium (NE)* serves as a more appropriate solution concept. Simultaneous-move security game models are particularly common for modeling interactions with terrorism, partially due to the fact that the defender’s actions are confidential in such settings (see, e.g., [Bier et al. 2007; Major 2002; Sandler et al. 2003, 2005]). In networked information systems, the interaction between the defender (system protector) and attacker (malware) is usually modeled as a simultaneous-move security game as well since malwares typically do not analyze past system behaviors. The goal is usually to compute some particular (e.g., best or worst) Nash equilibrium [Mavronicolas et al. 2006, 2005].

3.3. Security Games & Combinatorial Optimization

The main theme of this paper is to build connections between combinatorial optimization and equilibrium computation in security games. In particular, we consider the following combinatorial problem.

PROBLEM 3.2 (DEFENDER BEST RESPONSE (DBR)). *For any non-negative weight vector $\mathbf{w} \in \mathbb{R}_+^n$, compute*

$$\mathbf{e}^* = \arg \max_{\mathbf{e} \in \mathcal{E}} [\mathbf{w} \cdot \mathbf{e}].$$

The DBR problem over \mathcal{E} is to compute $\arg \max_{\mathbf{e} \in \mathcal{E}} [\mathbf{w} \cdot \mathbf{e}]$ for any input $\mathbf{w} \in \mathbb{R}_+^n$.

In other words, the DBR problem is to compute a defender pure strategy that maximizes the total weight it “collects”. We claim that Problem 3.2 is precisely the defender’s best response problem to an arbitrary attacker mixed strategy. To see this, given any attacker mixed strategy \mathbf{y} , we have $U^d(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^n x_i (y_i [r_i - c_i]) + \sum_{i=1}^n y_i c_i$

for any $\mathbf{x} \in \mathcal{P}$. Let $w_i = y_i[r_i - c_i] \geq 0$, then the defender’s best response to \mathbf{y} is $\arg \max_{\mathbf{x} \in \mathcal{P}} \mathbf{x} \cdot \mathbf{w} = \arg \max_{\mathbf{e} \in \mathcal{E}} \mathbf{e} \cdot \mathbf{w}$. Conversely, given any $\mathbf{w} \in \mathbb{R}_+^n$, it is easy to find an attacker mixed strategy $\mathbf{y} \in \Delta_n$ such that $y_i(r_i - c_i)$ is proportional to w_i for all $i \in [n]$, making Problem 3.2 equivalent to the defender’s best response to \mathbf{y} . Notice that the DBR problem is a combinatorial optimization problem over the set system \mathcal{E} . The difference among various security game models essentially lies in the structure of \mathcal{E} . Next, we illustrate how some typical DBR problems relate to standard combinatorial problems.

Uniform Matroid. In simple security settings, the defender has a certain number of security resources, say k resources; each resource can be assigned to protect any (one) target – i.e., no allocation constraints. Therefore, any subset of $[n]$ of size at most k is a defender pure strategy. In this case, \mathcal{E} is a uniform matroid and the DBR problem is simply to find the largest k weights. The deployed security system for LAX airport checkpoint placement – one of the earliest applications of security games – is captured by this model [Pita et al. 2008].

Bipartite Matching. A natural generalization of the uniform matroid case is that the resource allocation is constrained. In this case, the defender has k heterogeneous resources, and each resource can only be assigned to some particular targets associated with that resource. This naturally models several types of scheduling constraints in practice. For example, due to geographic constraints, policemen from a certain police station can only patrol the area around that station. Also, different types of security forces specialize in protecting different types of targets. Here, feasible assignments can be modeled as edges of a bipartite graph with security resources on one side and targets on another side. A defender pure strategy corresponds to a bipartite matching, and the DBR problem is to compute the maximum weighted bipartite matching.

Coverage Problems. In some domains, one security resource can cover several targets. One deployed real-world example is the federal air marshal scheduling problem, where one air marshal is assigned to protect several flights, but constrained on that the arrival destination of any former flight should be the starting point of the next flight [Tsai et al. 2009]. In other words, each security resource (i.e., air marshal) can protect a subset of targets (i.e., flights). Therefore, each pure strategy is the union of targets covered by each security resource. The DBR problem in this case is the maximum weighted coverage problem. Another natural example is to protect targets distributed on the plane and each security guard can cover a region of certain size. The DBR problem here is a 2-dimensional geometric maximum coverage problem.⁴ Other examples include patrolling on a graph (e.g., street map or network systems) in which a patroller at a node can protect all adjacent edges or a patroller on an edge can protect its two end nodes. The DBR problem here is the vertex or edge coverage problem.

Min-Cost Flow. Many security games are played out in *spatio-temporal* spaces. For example, the deployed security system in [Fang et al. 2013] helps to schedule the patrol boats of the US Coast Guard to protect the (moving) Staten Island ferries every day. Wildlife protection is another example [Fang et al. 2016]. One common way to handle such settings is to discretize the space and build a 2-D – spatial and temporal dimension – grid, and patrol the discrete (*space, time*) points (see Figure 1). However, starting from a position at time t , the positions that a security resource can possibly reach at time $t + 1$ are restricted due to various constraints like speed limit, terrain barriers, etc. For example, the move highlighted by red in Figure 1 may be infeasible since a patroller can not move that far within a small time period due to speed limit, while the blue-colored moves are feasible. This can be modeled by adding edges between time layers to indicate feasible moves. The patrolling schedule for each security

⁴For more information about geometric coverage, see the thesis [Leeuwen et al. 2009] and references therein.

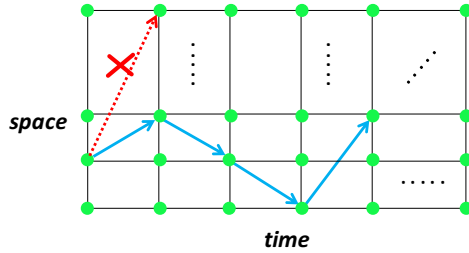


Fig. 1: Feasible (blue) and infeasible (red) moves in spatio-temporal spaces.

resource corresponds to a path across all time layers, which specifies the position this resource covers at each time point (see the blue path in Figure 1). The DBR problem is, for any given non-negative weights at each $(space, time)$ point, computing k paths for the k resources to maximize the total weights they cover. This can be converted to a min-cost max flow problem (with negative costs).

Packing Problems. Our last example is motivated by recent work on optimizing the allocation of security resources for passenger screening for the Transportation Security Administration (TSA) [Brown et al. 2016]. Consider an airport with n flights and flight i has m_i passengers. The TSA has several screening *tools*, e.g., x-ray, walk-through metal detector, chemicals, etc., and each screening tool has a capacity of the maximum number of passengers it can check. In contrast to the case of coverage problems where each resource can protect several targets, here several tools are required to screen one passenger. More precisely, each passenger is screened by a screening *team* which is a combination of several screening tools. By protecting the flight from a certain passenger we mean a passenger of that flight is screened, and identified as an attacker if he is, by a screening team.⁵ The DBR problem is to, given non-negative weight w_i for passengers in flight i , allocate as many passengers as possible to teams for screening, subject to each screening tool’s capacity constraint, so that the total weight of screened passengers are maximized. This is a very general packing problem. In fact, the reader may easily verify that even when $w_i = 1$ and $m_i = 1$ for any flight i , the problem encodes the independent set problem.

Remark 3.3. We note that the list of examples above is by no means complete – there are various other security settings with other combinatorial structures. For example, there is also study on budget constraints for acquiring security resources (e.g., [Bhattacharya et al. 2011]), which induces the budgeted version of the above combinatorial problems. In fact, real domains are usually more complicated with various types of constraints, involving intersections of these combinatorial structures.

4. SOLVING ZERO-SUM SECURITY GAMES IS A COMBINATORIAL PROBLEM

In this section, we focus on zero-sum security games. Recall that we use n to denote the total number of targets, and reward r_i [cost c_i] to denote the defender’s utility of covering [uncovering] target i when it is attacked. The defender seeks to maximize her utility while the attacker seeks to minimize the defender’s utility. We are interested in computing the minimax equilibrium for security games over \mathcal{E} in $\text{poly}(n)$ time. More

⁵In [Brown et al. 2016], each screening team has an effectiveness factor denoting the probability a team can identify an attacker. The setting here is slightly simplified with perfect effectiveness factor 1. Nevertheless, it still captures the core difficulty of the problem.

$$\begin{array}{ll}
\max & u \\
\text{s.t.} & x_i r_i + (1 - x_i) c_i \geq u, \text{ for } i = 1, 2, \dots, n. \\
& \sum_{\mathbf{e} \in \mathcal{E}} p_{\mathbf{e}} \cdot \mathbf{e} = \mathbf{x} \\
& \sum_{\mathbf{e} \in \mathcal{E}} p_{\mathbf{e}} = 1 \\
& p_{\mathbf{e}} \geq 0, \quad \text{for } \mathbf{e} \in \mathcal{E}.
\end{array}
\qquad
\begin{array}{ll}
\min & \sum_{i=1}^n c_i y_i + r \\
\text{s.t.} & r - \mathbf{e} \cdot \mathbf{w} \geq 0, \quad \text{for } \mathbf{e} \in \mathcal{E}. \\
& (r_i - c_i) y_i = w_i, \text{ for } i = 1, 2, \dots, n. \\
& \sum_{i=1}^n y_i = 1 \\
& y_i \geq 0, \quad \text{for } i = 1, 2, \dots, n.
\end{array}$$

Fig. 2: The defender’s maximin strategy

Fig. 3: The attacker’s minimax strategy

specifically, we seek to understand how the computational complexity of the minimax equilibrium relates to the complexity of the DBR problem. By convention, we sometimes call an algorithm for solving the DBR problem a *DBR oracle*. We prove the following equivalence theorem.

THEOREM 4.1. *There is a poly(n) time algorithm to compute the minimax equilibrium for zero-sum security games over \mathcal{E} , if and only if there is a poly(n) time algorithm to solve the DBR problem over \mathcal{E} .*

Since security games are bilinear games, it is not hard to see that the minimax equilibrium can be computed in polynomial time with access to an efficient DBR oracle. The reduction follows a standard primal-dual argument. In particular, Figure 2 and 3 exhibit the linear programs for computing the defender’s maximin strategy and the attacker’s minimax strategy. They are primal and dual of each other. Moreover, the separation oracle for the attacker’s linear program is precisely the DBR problem, thus both programs can be solved efficiently with access to an efficient DBR oracle (Theorem 2.1). We omit the details here.

What is interesting, however, is the reverse direction – i.e., solving the DBR problem is no harder than solving zero-sum security games. The minimax equilibrium, as an optimization problem, has a special objective function. It is not clear that such a special objective can be as hard as optimizing an arbitrary non-negative objective over \mathcal{E} . Moreover, security games are very special bilinear games: (i) the attacker’s mixed strategy set is a simplex; (ii) the defender’s payoff matrix is non-negative and diagonal; (iii) the attacker’s payoff matrix is non-positive and diagonal. Thus unsurprisingly, there have been attempts in the literature on solving security games without going through the DBR problem. Indeed, various other techniques have been employed to tackle security games, e.g., generalized Birkhoff-von Neumann theorem [Budish et al. 2013], techniques from convex and non-convex optimization, etc. However, the message conveyed in Theorem 4.1 is that, to solve security games in polynomial time, it is not only sufficient, but also necessary, to solve the DBR problem in polynomial time. We now provide a formal reduction from the DBR problem to minimax equilibrium computation.

Reducing DBR to Minimax Equilibrium

Recall that all the feasible marginal coverage probabilities form the polytope \mathcal{P} , and the vertices of \mathcal{P} form the set \mathcal{E} of defender pure strategies. We start with a simple fact of linear programming, which states that any linear program achieves optimality at some vertex of its feasible region (if non-empty).

FACT 4.2. $\max_{\mathbf{e} \in \mathcal{E}} [\mathbf{w} \cdot \mathbf{e}] = \max_{\mathbf{x} \in \mathcal{P}} [\mathbf{w} \cdot \mathbf{x}]$, where \mathcal{P} is the convex hull of \mathcal{E} .

By Fact 4.2, solving the DBR problem is equivalent to solving the linear program $\max_{\mathbf{x} \in \mathcal{P}} [\mathbf{w} \cdot \mathbf{x}]$. The main technical step of our proof is to reduce the membership checking problem for polytope \mathcal{P} to the computation of minimax equilibrium. Therefore, computing the minimax equilibrium will allow us to find a poly(n) time membership

oracle for \mathcal{P} . By the polynomial-time equivalence between membership checking and optimization (Theorem 2.2), we can conclude that the DBR problem can also be solved in $\text{poly}(n)$ time. Unfortunately, it turns out that membership checking for polytope \mathcal{P} cannot be easily reduced to the computation of minimax equilibrium because some $\mathbf{x} \in \mathcal{P}$ are entry-wise dominated by (i.e., entry-wise *less* than) other $\mathbf{x}' \in \mathcal{P}$, so that \mathbf{x} can never be a defender equilibrium strategy. To overcome this barrier, we first relax the polytope \mathcal{P} and work on a broader set $\widehat{\mathcal{P}}$ of marginal probabilities. We then make use of the non-negativity of weights in the DBR problem to transfer linear optimization over $\widehat{\mathcal{P}}$ back to linear optimization over \mathcal{P} .

We first introduce some notations. We call $\widehat{\mathbf{e}} \in \{0, 1\}^n$ a *sub pure strategy* (of \mathbf{e}) if there exists an $\mathbf{e} \in \mathcal{E}$ such that $\widehat{e}_i \leq e_i$ for all $i \in [n]$. Equivalently, the covered targets of a sub pure strategy $\widehat{\mathbf{e}}$ is a subset of the covered targets of some real pure strategy \mathbf{e} . For example, the all zero vector $\mathbf{0}$ is always a sub pure strategy; any pure strategy $\mathbf{e} \in \mathcal{E}$ is also a sub pure strategy (of itself). Notice that sub pure strategies are not necessarily feasible. For example, in the air marshal scheduling problem, a feasible schedule for an air marshal has to be a round trip, not a one-way flight. We now define the *relaxed* set of pure strategies

$$\widehat{\mathcal{E}} = \{ \text{all } \mathbf{e} \in \mathcal{E} \text{ and all their sub pure strategies } \}$$

to be the set of \mathcal{E} augmented with all sub pure strategies. The following lemma shows that, when considering the DBR problem, relaxing \mathcal{E} to $\widehat{\mathcal{E}}$ is without loss.

LEMMA 4.3. $\max_{\mathbf{e} \in \mathcal{E}}[\mathbf{e} \cdot \mathbf{w}] = \max_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}}[\widehat{\mathbf{e}} \cdot \mathbf{w}]$ for any $\mathbf{w} \in \mathbb{R}_+^n$.

PROOF. Since $\mathcal{E} \subseteq \widehat{\mathcal{E}}$, we have $\max_{\mathbf{e} \in \mathcal{E}}[\mathbf{e} \cdot \mathbf{w}] \leq \max_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}}[\widehat{\mathbf{e}} \cdot \mathbf{w}]$. To prove another direction, let $\widehat{\mathbf{e}}^* = \arg \max_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}}[\widehat{\mathbf{e}} \cdot \mathbf{w}]$. By definition, there exists $\mathbf{e}^* \in \mathcal{E}$ such that $\widehat{\mathbf{e}}^*$ is a sub pure strategy of \mathbf{e}^* . Since $\mathbf{w} \in \mathbb{R}_+^n$ is non-negative, we must have $\mathbf{e}^* \cdot \mathbf{w} \geq \widehat{\mathbf{e}}^* \cdot \mathbf{w}$. Therefore, $\max_{\mathbf{e} \in \mathcal{E}}[\mathbf{e} \cdot \mathbf{w}] \geq \mathbf{e}^* \cdot \mathbf{w} \geq \widehat{\mathbf{e}}^* \cdot \mathbf{w} = \max_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}}[\widehat{\mathbf{e}} \cdot \mathbf{w}]$, which concludes the proof. \square

We now define the *relaxed* polytope of marginal probabilities as follows.

$$\widehat{\mathcal{P}} = \{ \mathbf{x} : \mathbf{x} = \sum_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}} p_{\widehat{\mathbf{e}}} \cdot \widehat{\mathbf{e}}, \forall \mathbf{p} \in \Delta_{|\widehat{\mathcal{E}}|} \}.$$

We next show that the DBR problem over \mathcal{E} can be reduced to linear optimization (or equivalently, membership checking, due to Theorem 2.2) of this relaxed polytope $\widehat{\mathcal{P}}$. It is easy to show that the optimal objective value of the DBR problem is equal to $\max_{\mathbf{x} \in \widehat{\mathcal{P}}}[\mathbf{x} \cdot \mathbf{w}]$. However, the problem is that the optimal (vertex) solution to $\max_{\mathbf{x} \in \widehat{\mathcal{P}}}[\mathbf{x} \cdot \mathbf{w}]$, i.e., some $\widehat{\mathbf{e}}^* \in \widehat{\mathcal{E}}$, is a sub pure strategy, which may not be a feasible real pure strategy (since \mathbf{w} may have zero-valued entries), but an algorithm for the DBR problem must return a feasible optimal solution. Nevertheless, this can be handled by properly regularizing the linear program $\max_{\mathbf{x} \in \widehat{\mathcal{P}}}[\mathbf{x} \cdot \mathbf{w}]$ to make sure that its optimal vertex solution is always feasible for the DBR problem over \mathcal{E} . We refer the reader to the full version for a formal argument.

LEMMA 4.4. *The DBR problem over \mathcal{E} reduces to membership checking for polytope $\widehat{\mathcal{P}}$ in $\text{poly}(n)$ time.*

We are now ready to complete the reduction by reducing membership checking for polytope $\widehat{\mathcal{P}}$ to solving zero-sum security games over \mathcal{E} . Before that, we first show the following down-monotone property of the relaxed polytope $\widehat{\mathcal{P}}$, which turns out to be

crucial for the reduction. Lemma 4.5 follows from our construction of $\widehat{\mathcal{E}}$ which is a downward closed set system.

LEMMA 4.5. *Polytope $\widehat{\mathcal{P}}$ is down-monotone in \mathbb{R}_+^n . That is, for any $\mathbf{x} \in \widehat{\mathcal{P}}$ and $\mathbf{0} \leq \mathbf{x}' \leq \mathbf{x}$ (entry-wise), we have $\mathbf{x}' \in \widehat{\mathcal{P}}$.*

LEMMA 4.6. *Membership checking for polytope $\widehat{\mathcal{P}}$ reduces in $\text{poly}(n)$ time to computing the game value of zero-sum security games over \mathcal{E} .*

PROOF. Given any $\mathbf{x} \in \mathbb{R}^n$, we show how to check whether $\mathbf{x} \in \widehat{\mathcal{P}}$ by solving properly constructed zero-sum security games. We assume \mathbf{x} lies in \mathbb{R}_+^n since otherwise $\mathbf{x} \notin \widehat{\mathcal{P}}$. Given any $\mathbf{x} \in \mathbb{R}_+^n$, we construct the following security game instance:

- For any $i \in [n]$ such that $x_i = 0$, let $c_i = 1$ and $r_i = 2$. Therefore, $x_i r_i + (1 - x_i) c_i = 1$.
- For any $i \in [n]$ such that $x_i > 0$, let $c_i = 0$ and $r_i = \frac{1}{x_i}$. Therefore, $x_i r_i + (1 - x_i) c_i = 1$.

Note that in both cases, the condition $r_i > c_i$ is satisfied. We claim that the defender's optimal utility, i.e., the game value, in the above zero-sum security game is at least 1 if and only if $\mathbf{x} \in \widehat{\mathcal{P}}$.

\Rightarrow direction: Let $(\mathbf{p}^*, \mathbf{x}^*, u^*)$ be the optimal solution to the LP in Figure 2. If the defender's optimal utility is at least 1, i.e., $u^* \geq 1$. We have

$$x_i^* r_i + (1 - x_i^*) c_i \geq u^* \geq 1 = x_i r_i + (1 - x_i) c_i, \forall i \in [n].$$

Since $r_i > c_i$, this induces $x_i^* \geq x_i$ for any i . Notice that the optimal (feasible) marginal probability \mathbf{x}^* is in \mathcal{P} by definition and $\mathcal{P} \subseteq \widehat{\mathcal{P}}$, therefore $\mathbf{x}^* \in \widehat{\mathcal{P}}$. However, $x_i \leq x_i^*$ for any i and $\widehat{\mathcal{P}}$ is down monotone, so $\mathbf{x} \in \widehat{\mathcal{P}}$, as desired.

\Leftarrow direction: Let $\mathbf{x} \in \widehat{\mathcal{P}}$. So there exists $\mathbf{p} \in \Delta_{|\widehat{\mathcal{E}}|}$ such that

$$\mathbf{x} = \sum_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}} p_{\widehat{\mathbf{e}}} \cdot \widehat{\mathbf{e}}. \quad (4)$$

Notice that each $\widehat{\mathbf{e}}$ is a sub pure strategy of some real pure strategy $\mathbf{e}(\widehat{\mathbf{e}}) \in \mathcal{E}$. By substituting the $\widehat{\mathbf{e}}$ in Equation (4) by $\mathbf{e}(\widehat{\mathbf{e}})$, we have

$$\mathbf{x} = \sum_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}} p_{\widehat{\mathbf{e}}} \cdot \widehat{\mathbf{e}} \leq \sum_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}} p_{\widehat{\mathbf{e}}} \cdot \mathbf{e}(\widehat{\mathbf{e}}).$$

Since $\mathbf{x}' = \sum_{\widehat{\mathbf{e}} \in \widehat{\mathcal{E}}} p_{\widehat{\mathbf{e}}} \cdot \mathbf{e}(\widehat{\mathbf{e}})$ is a convex combination of pure strategies in \mathcal{E} , we have $\mathbf{x}' \in \mathcal{P}$. Moreover, by playing the mixed strategy with marginal probability \mathbf{x}' , the defender's expected utility is at least 1 since $x'_i \geq x_i$ for any $i \in [n]$. As a result, the optimal defender utility in this zero-sum game is at least 1, completing the proof. \square

Lemma 4.4 and 4.6 forms a reduction from the DBR problem to the minimax equilibrium. This together with the reduction from the minimax equilibrium to the DBR problem conclude the proof of Theorem 4.1.

4.1. When Gaming is Easier than Best Response

Security games are special bilinear games. Theorem 4.1 shows that the particular problem of computing the minimax equilibrium for zero-sum security games is as hard as the general DBR problem. A natural question is whether there are instances such that the hardness of gaming and best response are strictly separated. Notice, however, that the best response problem is essentially more general, thus no easier, than solving the game. Therefore, the question really is, whether there are instances where gaming

is easier than best response. The following proposition answers this in the affirmative for security games. Notice that this proposition does not contradict Theorem 4.1, since the constructed security game instances here have further restricted payoffs, which make solving the game easy but still maintain the hardness of the best response.

PROPOSITION 4.7. *There exist zero-sum security games such that the minimax equilibrium can be computed in polynomial time but the DBR problem is NP-hard.*

5. GENERAL-SUM SECURITY GAMES

In this section, we consider general-sum security games. Recall that such a game \mathcal{G} is given by a tuple $(\mathbf{r}, \mathbf{c}, \rho, \zeta, \mathcal{E})$ where r_i [c_i] is the defender’s reward [cost] and ρ_i [ζ_i] is the attacker’s reward [cost], if target i is attacked. We consider the computation of the two mostly adopted equilibrium concepts in the security game literature, namely, the strong Stackelberg equilibrium (SSE) and Nash equilibrium (NE). For each equilibrium concept, we prove analogous equivalence theorem as the zero-sum case.

THEOREM 5.1. *There is a $\text{poly}(n)$ time algorithm to compute the strong Stackelberg equilibrium for security games over \mathcal{E} , if and only if there is a $\text{poly}(n)$ time algorithm to solve the DBR problem over \mathcal{E} .*

PROOF. The “only if” direction follows from Theorem 4.1, the fact that the minimax equilibrium is payoff-equivalent to the strong Stackelberg equilibrium (SSE) in zero-sum games, and that zero-sum games are special cases of general-sum games. We prove the “if” direction. It is known that in normal-form two player games, the SSE can be computed by solving multiple linear programs [Conitzer and Sandholm 2006]. A similar linear program (LP) formulation can be derived for security games, except that each LP now has $\text{poly}(|\mathcal{E}|, n)$ many variables. Nevertheless, it can be shown that the separation problem for their dual programs is precisely the DBR problem, thus we can still solve each linear program efficiently with access to an efficient DBR oracle. We refer the reader to the full version for more details. \square

We now turn to the computation of Nash equilibria. As widely known in the literature of algorithmic game theory, computing one Nash equilibrium for two-player normal-form games is PPAD-hard [Chen et al. 2009; Daskalakis et al. 2009], and is only harder for general bilinear games [Garg et al. 2011]. Interestingly, it turns out that computing one Nash equilibrium in security games is relatively easy. This is due to the following characterization of Nash equilibria in security games by [Korzhyk et al. 2011b].

LEMMA 5.2. [Korzhyk et al. 2011b] *Consider a security game $\mathcal{G}(\mathbf{r}, \mathbf{c}, \rho, \zeta, \mathcal{E})$. Let $\bar{\mathcal{G}}(-\zeta, -\rho, \rho, \zeta, \mathcal{E})$ be the corresponding zero-sum security game by re-setting the defender’s utilities. Then (\mathbf{x}, \mathbf{y}) is a Nash equilibrium of \mathcal{G} if and only if $(\mathbf{x}, f(\mathbf{y}))$ is a minimax equilibrium of the zero-sum game $\bar{\mathcal{G}}$, where the one-to-one transform function $f : \mathbb{R}^n \rightarrow \mathbb{R}^n$ is defined as follows:*

$$f_i(\mathbf{y}) = \frac{1}{\lambda} \frac{r_i - c_i}{\rho_i - \zeta_i} y_i, \quad \forall i \in [n], \quad \text{where } \lambda = \sum_{i=1}^n \frac{r_i - c_i}{\rho_i - \zeta_i} y_i \text{ is the normalization factor.} \quad (5)$$

Moreover, Nash equilibria of \mathcal{G} are interchangeable. That is, if (\mathbf{x}, \mathbf{y}) and $(\mathbf{x}', \mathbf{y}')$ are both Nash equilibria, so are $(\mathbf{x}, \mathbf{y}')$ and $(\mathbf{x}', \mathbf{y})$. The attacker derives the same utility in any Nash equilibrium of \mathcal{G} .

Notice that the transform function defined in Equation (5) is *non-linear* due to the normalization factor. We sketch the intuition of Lemma 5.2 here, while refer the reader to [Korzhyk et al. 2011b] for a formal proof. Note that the mapping f only re-weights

those *non-zero* y_i 's whose indexes correspond to attacker best responses, so $f(\mathbf{y})$ is also a best response to \mathbf{x} in \mathcal{G} , thus also in $\bar{\mathcal{G}}$, since the defender strategy and *attacker* payoff structure in both games are the same. On the other hand, \mathbf{x} is a best response to \mathbf{y} in \mathcal{G} . From \mathcal{G} to $\bar{\mathcal{G}}$, the defender's utility on each target is changed. The idea here is to properly rescale the attacker's attacking probability to compensate for the defender's utility change so that the defender's best response does not change. The transform function f exactly does this. The interchangeability follows from the interchangeability of minimax equilibria of $\bar{\mathcal{G}}$. It is easy to see that the attacker's utility in any NE equals his (unique) utility in the zero-sum game $\bar{\mathcal{G}}$.

As a corollary of Lemma 5.2 and Theorem 4.1, there is a polynomial time algorithm to compute *one* Nash equilibrium for security games if and only if there is a polynomial time DBR oracle. However, it is widely known that the Nash equilibrium is not unique, so there are issues of equilibrium selection. In many security settings (e.g., [Mavronicolas et al. 2006, 2005]), the NE that maximizes or minimizes the defender's utility is a natural choice for analyzing the game. Though Lemma 5.2 shows that the attacker will derive the same utility in any NE, the defender's utilities are generally different in different NEs (examples are given in [Korzhyk et al. 2011b]).

As widely known, maximizing a player's utility over Nash equilibria is NP-hard even in two-player normal-form games [Conitzer and Sandholm 2008; Gilboa and Zemel 1989]. It will be appealing if the optimal NE can be efficiently computed in the security game, which is widely recognized as a successful application of game theory. Our next result shows that this is indeed the case! We prove that though the defender's equilibrium utility is generally a convex function of the attacker's mixed strategy, it becomes linear when restricted to the domain of the attacker *equilibrium strategies*. Therefore, maximizing or minimizing the defender's Nash equilibrium utility can both be efficiently handled.

THEOREM 5.3. *There is a $\text{poly}(n)$ time algorithm to compute the best and worst (for the defender) Nash equilibrium for security games over \mathcal{E} , if and only if there is a $\text{poly}(n)$ time algorithm to solve the DBR problem over \mathcal{E} . Here, by "best/worst" we mean the NE that maximizes/minimizes the defender's utility.*

PROOF. The "only if" direction follows from Theorem 4.1, the fact that all Nash equilibria are payoff equivalent to the minimax equilibrium in zero-sum games, and that zero-sum games are special cases of general-sum games. For the "if" direction, we only prove this for the case of computing the best Nash equilibrium since computing the worst Nash equilibrium is similar.

The defender's equilibrium utility is a function of the attacker's strategy, which we denote as $U_{NE}(\mathbf{y})$. We first derive the function form of $U_{NE}(\mathbf{y})$, as follows:

$$\begin{aligned} U_{NE}(\mathbf{y}) &= \max_{\mathbf{e} \in \mathcal{E}} U^d(\mathbf{e}, \mathbf{y}) \\ &= \max_{\mathbf{e} \in \mathcal{E}} \left[\sum_{i=1}^n e_i y_i (r_i - c_i) \right] + \sum_{i=1}^n y_i c_i. \end{aligned}$$

$U_{NE}(\mathbf{y})$ is a convex function since $\max_{\mathbf{e} \in \mathcal{E}} [\sum_{i=1}^n e_i y_i (r_i - c_i)]$ is convex. To compute the best equilibrium, we need to maximize this convex function over feasible \mathbf{y} which is generally difficult to handle. Interestingly, we show that the term $\max_{\mathbf{e} \in \mathcal{E}} [\sum_{i=1}^n e_i y_i (r_i - c_i)]$ becomes linear in \mathbf{y} when \mathbf{y} is restricted to be an attacker equilibrium strategy. Let (\mathbf{x}, \mathbf{y}) be an NE of \mathcal{G} . Recall from Lemma 5.2 that $(\mathbf{x}, f(\mathbf{y}))$ is the minimax equilibrium of the zero-sum game $\bar{\mathcal{G}}(-\zeta, -\rho, \rho, \zeta, \mathcal{E})$. Therefore,

$$\text{Val}(\bar{\mathcal{G}}) = U^d(\mathbf{x}, f(\mathbf{y})) = \max_{\mathbf{e} \in \mathcal{E}} \left[\sum_{i=1}^n f_i(\mathbf{y}) e_i (\rho_i - \zeta_i) \right] - \sum_{i=1}^n f_i(\mathbf{y}) \rho_i \quad (6)$$

is the game value of $\bar{\mathcal{G}}$. Recalling the transform $f_i(\mathbf{y}) = \frac{1}{\lambda} \frac{r_i - c_i}{\rho_i - \zeta_i} y_i$ and utilizing Equation (6), we have

$$\begin{aligned}
\max_{\mathbf{e} \in \mathcal{E}} \left[\sum_{i=1}^n e_i y_i (r_i - c_i) \right] &= \max_{\mathbf{e} \in \mathcal{E}} \left[\sum_{i=1}^n e_i \lambda f_i(\mathbf{y}) (\rho_i - \zeta_i) \right] && \text{By definition of } f_i(\mathbf{y}) \\
&= \lambda \cdot \max_{\mathbf{e} \in \mathcal{E}} \left[\sum_{i=1}^n e_i f_i(\mathbf{y}) (\rho_i - \zeta_i) \right] \\
&= \lambda \cdot \left[\text{Val}(\bar{\mathcal{G}}) + \sum_{i=1}^n f_i(\mathbf{y}) \rho_i \right] && \text{By Equation (6)} \\
&= \lambda \cdot \text{Val}(\bar{\mathcal{G}}) + \sum_{i=1}^n \frac{r_i - c_i}{\rho_i - \zeta_i} y_i \rho_i && \text{By definition of } f_i(\mathbf{y})
\end{aligned}$$

Crucially, $\text{Val}(\bar{\mathcal{G}})$ is a constant and does not depend on \mathbf{y} . Since $\lambda = \sum_{i=1}^n \frac{r_i - c_i}{\rho_i - \zeta_i} y_i$ is linear in \mathbf{y} , so is $\max_{\mathbf{e} \in \mathcal{E}} \left[\sum_{i=1}^n e_i y_i (r_i - c_i) \right]$. Therefore, $U_{NE}(\mathbf{y})$ is linear in \mathbf{y} . We note again that this is true only when \mathbf{y} is restricted to be an attacker NE strategy since the above derivation is not valid otherwise. Fortunately, we only need to optimize over Nash equilibria. Notice that $\text{Val}(\bar{\mathcal{G}})$ can be computed in $\text{poly}(n)$ time if the DBR admits a $\text{poly}(n)$ time algorithm (by Theorem 4.1). As a result, $U_{NE}(\mathbf{y})$ is a linear function of \mathbf{y} which can be evaluated efficiently. Therefore, computing the best Nash equilibrium reduces to a linear optimization over the set of all the attacker's Nash equilibrium strategies, denoted as Y_{NE} . To complete the proof, we now show that Y_{NE} is a polytope, and admits a $\text{poly}(n)$ time separation oracle. Therefore, by Theorem 2.1, we conclude that the best (for the defender) Nash equilibrium can be computed in $\text{poly}(n)$ time.

LEMMA 5.4. *The set of the attacker's Nash equilibrium strategies Y_{NE} is a polytope, and admits a $\text{poly}(n)$ time separation oracle if the DBR problem over \mathcal{E} can be solved in $\text{poly}(n)$ time.*

Proof of Lemma 5.4. First, compute an arbitrary Nash equilibrium $(\tilde{\mathbf{x}}, \tilde{\mathbf{y}})$ of \mathcal{G} . By Lemma 5.2 and Theorem 4.1, this can be done in $\text{poly}(n)$ time. We claim that the set of attacker's Nash equilibrium strategies Y_{NE} is characterized precisely by the following three sets of linear constraints on $\mathbf{y} \in \mathbb{R}^n$:

$$\sum_{i=1}^n y_i (\rho_i \cdot [1 - \tilde{x}_i] + \zeta_i \cdot \tilde{x}_i) \geq \rho_k \cdot [1 - \tilde{x}_k] + \zeta_k \cdot \tilde{x}_k, \quad \forall k \in [n]. \quad (7)$$

$$\sum_{i=1}^n y_i (r_i \cdot \tilde{x}_i + c_i \cdot [1 - \tilde{x}_i]) \geq \sum_{i=1}^n y_i (r_i \cdot e_i + c_i \cdot [1 - e_i]), \quad \forall \mathbf{e} \in \mathcal{E}. \quad (8)$$

$$\sum_{i=1}^n y_i = 1 \quad \text{and} \quad y_i \geq 0, \quad \forall i \in [n]. \quad (9)$$

Inequality (7) restricts \mathbf{y} to be an attacker best response to the defender equilibrium strategy $\tilde{\mathbf{x}}$; Inequality (8) means that $\tilde{\mathbf{x}}$ should be a defender best response to the attacker mixed strategy \mathbf{y} .

We first show that these constraints are necessary. By Lemma 5.2, Nash equilibria in security games are interchangeable. Therefore if \mathbf{y} is an attacker equilibrium strategy, $(\tilde{\mathbf{x}}, \mathbf{y})$ is a Nash equilibrium. Thus \mathbf{y} and $\tilde{\mathbf{x}}$ are best responses to each other, as described by Inequality (7) and (8). Since \mathbf{y} is a mixed strategy, thus Constraint (9) holds.

To show that these constraints are also sufficient, let \mathbf{y} be any vector that satisfies these constraints. Constraint (9) restricts \mathbf{y} to be a valid mixed strategy; Inequality (7) and (8) induce that $(\tilde{\mathbf{x}}, \mathbf{y})$ is a Nash equilibrium. To sum up, Constraints (7) ~ (9) precisely characterize the set Y_{NE} .

We now show that Y_{NE} admits a $\text{poly}(n)$ time separation oracle. Notice that Constraints (7) and (9) can be explicitly checked in $\text{poly}(n)$ time. Constraint (8) can be simplified as follows:

$$\begin{aligned} \sum_{i=1}^n y_i (r_i \cdot \tilde{x}_i + c_i \cdot [1 - \tilde{x}_i]) &\geq \sum_{i=1}^n y_i (r_i \cdot e_i + c_i \cdot [1 - e_i]), & \forall \mathbf{e} \in \mathcal{E}. \\ \Leftrightarrow \sum_{i=1}^n y_i (r_i - c_i) \cdot \tilde{x}_i &\geq \sum_{i=1}^n y_i (r_i - c_i) \cdot e_i, & \forall \mathbf{e} \in \mathcal{E}. \end{aligned}$$

The inequality can be checked by feeding $w_i = y_i (r_i - c_i)$ into the DBR problem over \mathcal{E} and examining whether the optimal objective value is at most $\sum_{i=1}^n y_i (r_i - c_i) \cdot \tilde{x}_i$, and if not, returning the optimal \mathbf{e} from the DBR problem as a certificate of the violated constraint. Therefore, a $\text{poly}(n)$ time DBR oracle yields a $\text{poly}(n)$ time separation oracle for Y_{NE} . This completes our proof of Lemma 5.2, as well as the proof of the theorem. \square

The following is a simple corollary of Theorem 5.3. It follows from the fact proved in Theorem 5.3 that the defender's equilibrium utility is a linear function over the set Y_{NE} of attacker equilibrium strategies and Y_{NE} is a polytope. Therefore, we can convert the computation of an NE with particular defender equilibrium utility to feasibility checking of a linear system.

COROLLARY 5.5. *Let \mathcal{G} be any security game, $U_{\max} [U_{\min}]$ be the best [worst] defender utility among all Nash equilibria. Then for any $U \in [U_{\min}, U_{\max}]$, there exists an NE of \mathcal{G} with defender utility U . Moreover, such an NE can be computed in polynomial time if the DBR problem over \mathcal{E} can be solved in polynomial time.*

6. CONSEQUENCES OF THE EQUIVALENCE THEOREMS

In this section we discuss some implications of these equivalence theorems. The following corollary of Theorem 4.1, 5.1 and 5.3 shows that the complexity of a security game is fully determined by the set system \mathcal{E} .

COROLLARY 6.1. *For any set system \mathcal{E} , the following problems reduce to each other in polynomial time:*

- (1) *Combinatorial optimization over \mathcal{E} for non-negative linear objectives;*
- (2) *Solving zero-sum security games over \mathcal{E} ;*
- (3) *Computing the strong Stackelberg equilibrium for security games over \mathcal{E} ;*
- (4) *Computing the best or worst (for the defender) Nash equilibrium for security games over \mathcal{E} .*

Corollary 6.1 provides a more convenient way to understand the computational complexity of security games, since the complexity of the combinatorial optimization problem over \mathcal{E} is much easier to study and analyze. In fact, Corollary 6.1 simultaneously implies the computational complexity for solving various types of security games. For example, when \mathcal{E} is any matroid set system or when the optimization over \mathcal{E} has a min-cost max flow formulation, the equilibrium of the security game can be computed efficiently. On the other hand, when the optimization over \mathcal{E} is a packing problem, a Knapsack problem or a coverage problem (even vertex coverage), the equilibrium computation for these security games is NP-hard in general.

Finally, using the combinatorial characterization of security games, we can easily recover and strengthen some known complexity results in the literature of security games, as well as resolve some open problems from previous work. For example, [Xu et al. 2014] considered the computation of the minimax equilibrium in zero-sum spatio-temporal security games. They proved that the DBR problem there is NP-hard, but the computation of the minimax equilibrium is left open. Theorem 4.1 resolves this open question. [Brown et al. 2016] studied the airport passenger screening game (see Section 3.3, the example of packing problems), and proved the NP-hardness to solve the game. This also follows from Corollary 6.1 and the fact that the independent set problem is NP-hard. [Gan et al. 2015] considered security games on graphs where targets are vertices. The defender chooses a subset of vertices to patrol, by which the patrolled vertices as well as their adjacent vertices are covered. The DBR problem is to, given a non-negative weight for each vertex, compute a subset of vertices so that the total weight of the covered vertices is maximized. This is the optimization variant of the dominating set problem, a well-known NP-hard problem. Therefore, not only computing the SSE is NP-hard (as shown by Gan et al.), our results imply that computing the minimax equilibrium is also NP-hard. [Korzhyk et al. 2010] considered security games in which the DBR problem is the coverage problem as discussed in Section 3.3. They showed polynomial solvability of security games when each (homogeneous) resource can protect a subset of at most 2 targets (e.g., a pair of round-trip flights). This also follows from Theorem 4.1 and the fact that weighted 2-cover is polynomial time solvable. The NP-hardness for the case with sets of at most 3 targets follows from Theorem 4.1 and the fact that 3-cover is NP-hard. Finally, [Letchford and Conitzer 2013] proved complexity results for security games played on graphs, where vertices are targets and each security resource can patrol a set of vertices on a path of the graph. Some of their results (e.g., the two positive results in the paper) can be easily recovered under our framework as well. We omit further details here due to space limit.

7. CONCLUSIONS AND DISCUSSIONS

In this paper, we systematically studied the computational complexity of equilibrium computation in security games. Our main result is the polynomial time equivalence between computing the three mostly adopted equilibrium concepts in security games, namely, the minimax equilibrium, strong Stackelberg equilibrium, best/worst Nash equilibrium, and computing the defender’s best response. We believe that our results form a theoretical basis for further algorithm design and complexity analysis in security games.

Future research can take a number of directions. First, given that exactly solving the DBR problem is NP-hard in many cases, it is interesting to examine the approximate version of all our equivalence theorems. That is, how an approximate defender best response oracle relates to the approximate computation of an equilibrium. We note that using the no-regret learning framework, one can convert an FPTAS for the DBR problem to an algorithm for computing an ϵ -minimax equilibrium (see [Immorlica et al. 2011]), but the reverse direction and other generalizations are open. Our results on computing the best/worst Nash equilibrium are surprising. Since the security game is a special class of bilinear games, we wonder whether similar results hold in other special (and interpretable) class of bilinear games. Finally, there are several ways to generalize our model. For example, the players’ utility functions may not be linear, but can still be compactly represented and computed. One particular example is the network interdiction game played on a graph [Tsai et al. 2010; Washburn and Wood 1995], in which the defender chooses edges to defend and the attacker chooses a path to attack. The task of the defender is to interdict the attacker at a certain edge. This is not captured by our (bilinear) framework. Another generalization is to allow the attacker

to attack multiple targets [Korzhyk et al. 2011a]. We wonder how the computational complexity of the proposed four problems relates to each other in these generalized settings.

ACKNOWLEDGMENTS

We would like to thank Shaddin Dughmi, Milind Tambe and Vincent Conitzer for helpful discussions. This work is supported by NSF grant CCF- 1350900 and MURI grant W911NF-11-1-0332.

REFERENCES

- Bo An, Eric Shieh, Milind Tambe, Rong Yang, Craig Baldwin, Joseph DiRenzo, Ben Maule, and Garrett Meyer. 2012. PROTECT—A Deployed Game Theoretic System for Strategic Security Allocation for the United States Coast Guard. *AI Magazine* 33, 4 (2012), 96.
- Sayan Bhattacharya, Vincent Conitzer, and Kamesh Munagala. 2011. Approximation algorithm for security games with costly resources. In *Internet and Network Economics*. Springer, 13–24.
- Vicki Bier, Santiago Oliveros, and Larry Samuelson. 2007. Choosing what to protect: Strategic defensive allocation against an unknown attacker. *Journal of Public Economic Theory* 9, 4 (2007), 563–587.
- Matthew Brown, Arunesh Sinha, Aaron Schlenker, and Milind Tambe. 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)*.
- Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. 2013. Designing random allocation mechanisms: Theory and applications. *The American Economic Review* 103, 2 (2013), 585–623.
- Xi Chen, Xiaotie Deng, and Shang-Hua Teng. 2009. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)* 56, 3 (2009), 14.
- Vincent Conitzer and Tuomas Sandholm. 2006. Computing the optimal strategy to commit to. In *Proceedings of the 7th ACM EC*. ACM, 82–90.
- Vincent Conitzer and Tuomas Sandholm. 2008. New complexity results about Nash equilibria. *Games and Economic Behavior* 63, 2 (2008), 621–641.
- C. Daskalakis, P. W Goldberg, and C. Papadimitriou. 2009. The complexity of computing a Nash equilibrium. *SIAM J. Comput.* 39, 1 (2009), 195–259.
- Fei Fang, Albert Xin Jiang, and Milind Tambe. 2013. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *AAMAS 2013*.
- Fei Fang, Thanh H. Nguyen, Rob Pickles, Wai Y. Lam, Gopalasamy R. Clements, Bo An, Amandeep Singh, Milind Tambe, and Andrew Lemieux. 2016. Deploying PAWS: Field Optimization of the Protection Assistant for Wildlife Security. In *IAAI*.
- Joan Feigenbaum, Daphne Koller, and Peter Shor. 1995. A game-theoretic classification of interactive complexity classes. In *Structure in Complexity Theory Conference, 1995., Proceedings of Tenth Annual IEEE*. IEEE, 227–237.
- Jiarui Gan, Bo An, and Yevgeniy Vorobeychik. 2015. Security Games with Protection Externalities. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Jugal Garg, Albert Xin Jiang, and Ruta Mehta. 2011. Bilinear games: Polynomial time algorithms for rank based subclasses. In *WINE*. Springer, 399–407.
- Itzhak Gilboa and Eitan Zemel. 1989. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior* 1, 1 (1989), 80–93.
- Martin Grötschel, László Lovász, and Alexander Schrijver. 2012. *Geometric algorithms and combinatorial optimization*. Vol. 2. Springer Science & Business Media.
- Nicole Immorlica, Adam Tauman Kalai, Brendan Lucier, Ankur Moitra, Andrew

- Postlewaite, and Moshe Tennenholtz. 2011. Dueling algorithms. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*. ACM, 215–224.
- Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. 2010. Complexity of Computing Optimal Stackelberg Strategies in Security Resource Allocation Games.
- Dmytro Korzhyk, Vincent Conitzer, and Ronald Parr. 2011a. Security games with multiple attacker resources. In *IJCAI*.
- Dmytro Korzhyk, Zhengyu Yin, Christopher Kiekintveld, Vincent Conitzer, and Milind Tambe. 2011b. Stackelberg vs. Nash in Security Games: An Extended Investigation of Interchangeability, Equivalence, and Uniqueness. *J. Artif. Int. Res.* (2011).
- Van EJ Leeuwen and others. 2009. *Optimization and approximation on systems of geometric objects*.
- Joshua Letchford and Vincent Conitzer. 2013. Solving Security Games on Graphs via Marginal Probabilities. In *AAAI*.
- John A Major. 2002. Advanced techniques for modeling terrorism risk. *The Journal of Risk Finance* 4, 1 (2002), 15–24.
- M. Mavronicolas, L. Michael, V. Papadopoulou, A. Philippou, and P. Spirakis. 2006. The price of defense. In *Mathematical Foundations of Computer Science 2006*. Springer.
- Marios Mavronicolas, Vicky Papadopoulou, Anna Philippou, and Paul Spirakis. 2005. A graph-theoretic network security game. In *Internet and Network Economics*. Springer, 969–978.
- J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. 2008. Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles International Airport. In *AA-MAS: industrial track*.
- Todd Sandler and others. 2003. Terrorism & game theory. *Simulation & Gaming* 34, 3 (2003), 319–337.
- Todd Sandler and others. 2005. Counterterrorism a game-theoretic analysis. *Journal of conflict resolution* 49, 2 (2005), 183–200.
- Milind Tambe. 2011. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press.
- Jason Tsai, Shyamsunder Rathi, Christopher Kiekintveld, Fernando Ordonez, and Milind Tambe. 2009. IRIS - A Tool for Strategic Security Allocation in Transportation Networks. In *AAMAS - Industry Track*.
- J. Tsai, Z. Yin, Jun-Y. Kwak, D. Kempe, C. Kiekintveld, and M. Tambe. 2010. Urban security: Game-theoretic resource allocation in networked physical domains. In *AAAI*.
- Heinrich von Stackelberg. 1934. *Marktform und Gleichgewicht*. Springer, Vienna.
- Bernhard von Stengel and Shmuel Zamir. 2004. Leadership with commitment to mixed strategies. *CDAM Research Report LSE-CDAM-2004-01, London School of Economics* (2004).
- Alan Washburn and Kevin Wood. 1995. Two-person zero-sum games for network interdiction. *Operations Research* 43, 2 (1995), 243–251.
- Haifeng Xu, Fei Fang, Albert Xin Jiang, Vincent Conitzer, Shaddin Dughmi, and Milind Tambe. 2014. Solving Zero-Sum Security Games in Discretized Spatio-Temporal Domains. In *AAAI*.
- Z. Yin, A. X. Jiang, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. P Sullivan. 2012. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine* 33, 4 (2012), 59.