# The Mysteries of Security Games: Equilibrium Computation becomes Combinatorial Algorithm Design*

Haifeng Xu†
Department of Computer Science
University of Southern California
haifengx@usc.edu

## ABSTRACT

The *security game* is a basic model for resource allocation in adversarial environments. One player (i.e., the *defender*) allocates her limited resources to defend critical targets and an adversary (i.e., the *attacker*) seeks his most favorable target to attack. In the past decade, there has been a surge of research interests in computing, analyzing and refining security games that are motivated by applications from various real 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, US Coast Guard and Federal Air Marshal Service [30], as well as non-governmental organizations [11]. Among all these research and applications, equilibrium computation serves as a foundation.

In this paper, we study security games from a theoretical perspective and provide a unified view of various security game models. In particular, we show that 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 literature, including all the deployed systems, and beyond; The set system $\mathcal{E}$ arising from various settings 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. More precisely, 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 infers the hardness [polynomial solvability] of all the others. Here by "games over $\mathcal{E}$" we mean the class of security games with any valid payoff structures, but a fixed set $\mathcal{E}$ of defender pure strategies. This shows that the complexity of a security game is essentially captured by the set system $\mathcal{E}$. We view drawing these connections as an important conceptual contribution of this paper.

## Keywords

Security Games, Equilibrium Computation, Complexity

## 1. INTRODUCTION

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 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, US Coast Guard for port patrolling and the Federal Air Marshal Service for scheduling air marshals [30]; Recently, new models and algorithms have been tested, and in preparation for deployment, by non-governmental organizations in Malaysia for wildlife protection [11] and by the Transportation Security Administration for airport passenger screening [4].

Among all these research and applications, 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 [30] for a review). However, many of these algorithms are based on integer linear programs, the techniques of column generation and sometimes, heuristics, which may run in exponential time or output non-optimal solutions. The computational complexity for solving these games are 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 aim at a systematic study on the computational complexity of the three main equilibrium concepts adopted

*The full version of this paper is currently in submission to ACM EC'16. Compared with the full version, this version omits most of the formal proofs. A full version of the paper is also available at arXiv with the same title.

†Supported by MURI grant W911NF-11-1-0332 and NSF grant CCF- 1350900.

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

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 literature, and provide complexity analysis under 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 $\mathbf{x}^T A\mathbf{y} + \alpha \cdot \mathbf{y}$; 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 $\Delta_n$. Moreover, the matrix $A$ is a *non-negative* diagonal matrix for the defender, while a *non-positive* diagonal matrix for the attacker. Interestingly, the vertexes 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 literature, including all the deployed security systems, and beyond. 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 games with valid 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 infers the hardness [polynomial solvability] of all the others. This shows that, the complexity of a security game is essentially captured 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 former work.

The result of computing the best/worst Nash equilibrium comes as a surprise. As widely known, maximizing a player's utility over Nash equilibria is NP-hard even in two-player normal-form games [15, 8]. It is appealing that security games, which capture a broad class of important real-world applications, "escaped" from the intractability in many cases. For the zero-sum case, it is not surprising that the minimax equilibrium can be reduced to the defender's best response problem, i.e., optimization over $\mathcal{E}$. The interesting direction is the opposite. That is, the best response problem – a general combinatorial optimization problem – can be reduced to the computation of the minimax equilibrium, i.e., an optimization problem with very specific objective. This is surprising since security games have very simple payoff structures, and are far less general than bilinear games. Nevertheless, the equilibrium objective is still rich enough to capture a general optimization problem. As a complement to this result, we show that, if we *further* restrict the payoff structure of security games, there exist zero-sum security games in which the minimax equilibrium can be computed in polynomial time but the best response problem is NP-hard.

To prove these results, one of the main challenges is to propose universal reductions between these problems, since we are not working on any specific instance. Therefore, we have to treat all these problems as abstractly given. Our reductions make use of the polynomial time equivalence between linear optimization, separation and membership checking for polytopes.

## 1.2 Related Work

Several papers in the security game literature have examined the computational complexity of security games in particular settings. The most relevant are the following two papers: Korzhyk et al. [18] consider the security settings where each security resource can be allocated to protect a subset of targets; Letchford and Conitzer [22] consider security games on graphs where targets are nodes and security resources patrol along paths. They prove polynomial solvability or NP-hardness under different conditions. To 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., [13, 4]. We note that our framework only concerns the basic security game models. There are many refinements of the basic models, e.g., the Bayesian setting [26], repeated setting [36], stochastic setting [33], etc. Examining the complexity of these settings is an interesting future work, but is not the focus of the presented work.

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 [9, 6]. In the same setting, computing the Nash equilibrium that maximizes one player's utility is NP-hard [15, 8], but the Strong Stackelberg equilibrium can be computed in polynomial time by solving linear programs [7]. Immorlica et al. [17] 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 $\epsilon$-minimax equilibrium to an additive FPTAS of the player's best response oracle, using the no regret learning framework. However, they do not consider the other direction, namely, reduction from best response to equilibrium computation. Garg et al. [14] 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 (approximation) algorithms when the payoff matrices have low rank.

**Outline.** Section 2 reviews game theory basics. Section 3 formally describes the unified formulation of security games, equilibrium concepts adopted in literature and relation to various combinatorial problems. The reduction between the minimax equilibrium and defender's best response problem is given in Section 4, while the reduction between the Strong Stackelberg equilibrium [best/worst Nash equilibrium] and defender's best response problem is provided in Section 5. We discuss implications of our results in Section 6, and conclude in Section 7.

# 2. PRELIMINARIES

In this paper, we focus on two-player games. A normal-form two-player game is given by two matrices $A, B \in \mathbb{R}^{m\times n}$. Given mixed strategies $\mathbf{x} \in \Delta_m$ and $\mathbf{y} \in \Delta_n$ played by player 1 and 2 respectively, player 1 [player 2] derives expected utility $\mathbf{x}^T A\mathbf{y}$ [$\mathbf{x}^T B\mathbf{y}$]. Slightly generalizing the normal-form two-player games are the *bilinear* games (see, e.g., [17, 14]), in which, instead of simplexes, each player' mixed strategies lie in a general polytope.

Formally, a bilinear game is given by a pair of matrices $(A, B)$ and *action polytopes* $(\mathcal{P}, \mathcal{Q})$. Given that player 1 plays $\mathbf{x} \in \mathcal{P}$ and player 2 plays $\mathbf{y} \in \mathcal{Q}$, the utilities for player 1 and 2 are $\mathbf{x}^T A\mathbf{y}$ and $\mathbf{x}^T B\mathbf{y}$ respectively. As we will show in Section 3, security games are bilinear games with slightly richer payoff structures of the form $\mathbf{x}^T A\mathbf{y} + \alpha \cdot \mathbf{y}$ and $\mathbf{x}^T B\mathbf{y} + \beta \cdot \mathbf{y}$ for some vector $\alpha, \beta$. Note that each vertex of the polytope is a player pure strategy, and a player may have exponentially many pure strategies in a bilinear game even though her action polytope (e.g., a hypercube) can be compactly represented. Throughout this paper, we assume all the action polytopes are compact.

**Nash Equilibrium (NE).** A strategy profile $(\mathbf{x}, \mathbf{y})$ is called a Nash Equilibrium (NE), if

$$\mathbf{x}^T A\mathbf{y} \geq \mathbf{x}'^T A\mathbf{y},\ \forall \mathbf{x}' \in \mathcal{P} \quad \text{and} \quad \mathbf{x}^T B\mathbf{y} \geq \mathbf{x}^T B\mathbf{y}',\ \forall \mathbf{y}' \in \mathcal{Q}.$$

According to Nash's theorem, there exists at least one NE, possibly multiple NEs, in a bilinear game. As observed in [14], bilinear games encode many interesting classes of games including two-player normal-form games, two-player Bayesian games, polymatrix games, etc., therefore computing an NE is hard in general bilinear games.

**Strong Stackelberg Equilibrium (SSE).** The NE captures the equilibrium outcome of a simultaneous-move game. However, when a player, say player 1, can move before another player, the NE is not appropriate any more. In this setting, the Strong Stackelberg Equilibrium (SSE), which was originally introduced by Heinrich Freiherr von Stackelberg to capture market competition with leader and follower firms, serves as a more appropriate solution concept. A two-player Stackelberg game is played between a *leader* and a *follower*. The leader moves first, or equivalently, commits to a strategy; The follower observes the leader's strategy and best responds. The leader's optimal strategy, together with the follower's best response, forms an SSE. Formally, let $\mathbf{y}_\mathbf{x} = \arg\max_{\mathbf{y}' \in \mathcal{Q}} \mathbf{x}^T B\mathbf{y}'$ denote the follower's best response to a leader strategy $\mathbf{x} \in \mathcal{P}$. A strategy profile $(\mathbf{x}, \mathbf{y})$ is called a Strong Stackelberg Equilibrium (SSE), if

$$\mathbf{x} = \arg\max_{\mathbf{x}' \in \mathcal{P}} \mathbf{x}'^T A\mathbf{y}_{\mathbf{x}'}, \qquad \text{and} \qquad \mathbf{y} = \mathbf{y}_\mathbf{x}.$$

Without loss of generality, we can assume $\mathbf{y}$ is a pure strategy since it is a best response to $\mathbf{x}$.

**Zero-Sum Games and the Minimax Equilibrium.** When $A = -B$, the bilinear game is zero-sum (zero-sum security games also require $\alpha = -\beta$). It is well-known that in zero-sum games, all standard equilibrium concepts, including the NE and SSE, are payoff-equivalent to the minimax equilibrium. A strategy profile $(\mathbf{x}, \mathbf{y})$, where $\mathbf{x} \in \mathcal{P}$ and $\mathbf{y} \in \mathcal{Q}$, is called a minimax equilibrium if

$$\mathbf{x}^T A\mathbf{y} \geq \mathbf{x}'^T A\mathbf{y},\ \forall \mathbf{x}' \in \mathcal{P} \quad \text{and} \quad \mathbf{x}^T A\mathbf{y} \leq \mathbf{x}^T A\mathbf{y}',\ \forall \mathbf{y}' \in \mathcal{Q}.$$

If $(\mathbf{x}, \mathbf{y})$ is a minimax equilibrium, $\mathbf{x}$ is called player 1's *maximin* strategy, $\mathbf{y}$ is called player 2's *minimax* strategy and $V = \mathbf{x}^T A\mathbf{y} = \max_{\mathbf{x}' \in \mathcal{P}} \min_{\mathbf{y}' \in \mathcal{Q}} \mathbf{x}'^T A\mathbf{y}'$ is called the *value* of the game. Each zero-sum game has a unique game value.

# 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 aims 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. The defender possesses multiple security resources to be allocated. 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 (realistic) examples in Section 3.3.

A more convenient representation of a pure strategy, as will be used throughout this paper, is to use a binary vector $\mathbf{e} \in \{0, 1\}^n$, in which the entries of value 1 specify the covered targets. Note that $\mathbf{e}$ can also be interpreted as the marginal coverage probabilities of each target in this pure strategy.[1] 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 $n$, while *not* in $|\mathcal{E}|$. By convention, we will simply call the running time of an algorithm or the size of a program formulation *exponential* if it is polynomial in $|\mathcal{E}|$. A defender mixed strategy is a distribution $\mathbf{p}$ 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 $\mathbf{y} \in \Delta_n$ 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 $\zeta_i$ if target $i$ is covered or a reward $\rho_i$ if $i$ is uncovered.[2] 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.[3] We summarize 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 called* 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 all games with valid 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

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

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 $U^d(\mathbf{p}, \mathbf{y})$

[1] Throughout this paper, we will 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 standard.

[2] To help the reader to remember, the Greek letters $\rho, \zeta$ have similar appearances as $r, c$.

[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 the case here.

to denote the defender's expected utility, which can be expressed as

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

where

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

is the *marginal* coverage probability of target $i$. Let $\mathbf{x} = (x_1,...,x_n)^T$ denote the marginal probabilities of all targets induced by the mixed strategy $\mathbf{p}$. Notice that the marginal probabilities of a pure strategy $\mathbf{e}$ is precisely $\mathbf{e}$ itself. Equation (1) shows that the defender's utility can be compactly expressed using the marginal probabilities of a defender mixed strategy. Moreover, the expected defender utility has the form $\mathbf{x}^T A\mathbf{y} + \alpha \cdot \mathbf{y}$ for some *non-negative* diagonal matrix $A$, and is bilinear in $\mathbf{x}$ and $\mathbf{y}$. 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) marginals. 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 represented in the following form. We note that $U^a(\mathbf{x},\mathbf{y})$ has the form $\mathbf{x}^T B\mathbf{y} + \beta \cdot \mathbf{y}$ for some *non-positive* diagonal matrix $B$, and is bilinear in $\mathbf{x}$ and $\mathbf{y}$.

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

## 3.2 Equilibrium Concepts

Many security games, including some deployed security systems [1, 37], are modeled as zero-sum games. In this case, $r_i = -\zeta_i$ and $c_i = -\rho_i$ for all $i \in [n]$, i.e., the defender's reward [cost] is the negative of the attacker's cost [reward]. For example, in the deployed security-game system for patrolling proof-of-payment metro-systems [37], 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 when failing to catch the evader. In zero-sum games, all the standard equilibrium concepts are payoff-equivalent to the well-known *minimax equilibrium*, and our goal is to compute the minimax equilibrium in $poly(n)$ time.

When the game is *not* zero-sum, i.e., the defender and attacker have different values over targets, the main solution concept adopted in the literature of security games is the *Strong Stackelberg Equilibrium* (**SSE**) [30]. 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. 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.

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, sometimes even the attacker intends to do surveillance, he cannot observe the defender's strategies due to limited attacker resources and, sometimes, confidentiality of the defender's resource allocation (e.g., plainclothes police). In these settings, the defender cannot commit to a strategy,[4] 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., [23, 27, 28, 3]). 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 do not analyze system's history behaviors, and the goal is to compute some particular (e.g., best or worst) Nash equilibrium [25, 24].

## 3.3 Security Games & Combinatorial Optimization

One of the main themes of this paper is to build connections between combinatorial optimization and equilibrium computation in security games. We view drawing these connections as an important conceptual contribution of this paper. 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 weights 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 $U^d(\mathbf{x},\mathbf{y}) = \sum_{i=1}^n x_i \Big( y_i [r_i - c_i] \Big) + \sum_{i=1}^n y_i c_i$ for any $\mathbf{x} \in \mathcal{P}$. Let $w_i = y_i[r_i - c_i] \geq 0$. Since the term $\sum_{i=1}^n y_i c_i$ is not affected by the defender strategy, the defender's best response to $\mathbf{y}$ is $\arg\max_{\mathbf{x}\in\mathcal{P}} \mathbf{x}\cdot\mathbf{w} = \arg\max_{\mathbf{e}\in\mathcal{E}} \mathbf{e}\cdot\mathbf{w}$. Oppositely, 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 at the structure of $\mathcal{E}$. In the following context, 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., there are no allocation constraints. As a result, 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

[4]Technically, the defender can still commit to play some strategy. However, the attacker cannot observe or verify the defender's strategy, making the commitment not effective.

LAX airport security game system deployed in 2007 – one of the earliest applications of security games – is captured by this model [26].

**Bipartite Matching.** A natural generalization of the uniform matroid case is that the resource allocation has constraints. In this case, the defender has $k$ heterogeneous resources, and each resource can only be allocated 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. The feasibility constraints can be modeled as edges of a bipartite graph with security resources on one side and targets on another side; A resource can be assigned to a target if and only if there exists an edge between them in the bipartite graph. Any defender pure strategy corresponds to a bipartite matching and the DBR problem is to compute the maximum weighted bipartite matching.

**Coverage Problem.** In some domains, one security resource can cover several targets. One deployed real example is the scheduling problem of the federal air marshals, where one air marshal is scheduled to protect several flights, but constrained on that the arrival destination of any former flight should be the starting point of the next flight [31]. In other words, each security resource (i.e., air marshal) can protect a restricted subset of targets (i.e., flights). As a result, 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.[5]. Other examples include patrolling on a graph (e.g., street graph of a city or network systems) in which a patroller at a node can protect all the 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 both space and time, which is also referred as *spatio-temporal* security games. For example, the deployed security system in [10] helps to schedule the patrol boats of US Coast Guard to protect the (moving) Staten Island ferries during the day time. Conservation area patrolling for protecting wildlife is another example [11]. One common way to handle such settings is to discretize the space to build a 2-D – spatial and temporal dimension – grid, and patrol the discrete *(space, time)* points (see Figure 1). The constraint here is that 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 is infeasible since the patroller can not move to the end 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 resource corresponds to a path across all time layers, which specifies the target 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 solved by adding a super source and sink to the graph, and then computing a $k$-unit min-cost integer flow with negative costs.

**Packing.** Our last example is motivated by recent work to optimize the allocation of security resources for passenger screening

[5]For more information about geometric coverage, see the thesis [21] by Leeuwen and the references therein.

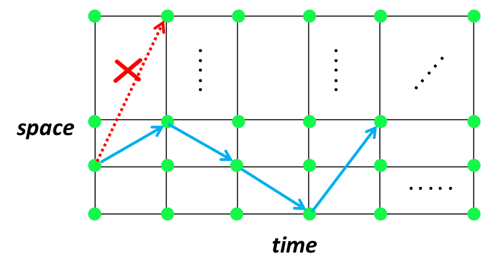


**Figure 1: Feasible (blue) and Infeasible (red) Moves in Discretized Spatio-Temporal Games.**

for the Transportation Security Administration (TSA) in United States [4]. 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. Opposite from the coverage case where each resource can protect several targets, here several tools are needed to screen one passenger. More precisely, each passenger is screened by a screening *team* which is a combination of several screening tools. By attacking flight $i$ we mean the attacker becomes a passenger for that flight (i.e., bought its ticket), and brings attack equipments with him. By protecting the flight from a certain passenger we mean that passenger is screened, and identified as an attacker if he is, by a screening team.[6] The DBR problem is to, given non-negative weight $w_i$ for all passengers in any flight $i$, allocate as many passengers as possible to teams for screening, subject to each screening tool's capacity constraint, so that the total weights of screened passengers are maximized. This is a very general packing problem. In fact, the reader may easily check that when $w_i = 1$ and $m_i = 1$ for any flight $i$, the problem encodes a vertex packing (i.e., independent set) problem.

REMARK 3.3. *We note that the examples above do not represent all the settings of security games. For example, there is also study on budget constraints for acquiring security resources (e.g., [2]), which induces the budgeted version of the above combinatorial problems (e.g., the uniform matroid case becomes a* Knapsack *problem). In fact, real domains are usually more complicated with various types of constraints, involving intersections of these combinatorial structures. Nevertheless, the combinatorial nature of these problems does not change and the DBR problem still has the same form as in Definition 3.1.*

# 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 $poly(n)$ time. More specifically, we seek to understand how the computational complexity of the minimax equilibrium relates to the complexity of the DBR problem. By conven-

[6]In [4], 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.

tion, 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 $\mathcal{E}$.*

Realizing that security games are bilinear games, it is not surprising that the minimax equilibrium can be computed in polynomial time with access to an efficient DBR oracle. In fact, it is not hard to check that even general bilinear zero-sum games can be solved with access to efficient best response oracles. Our reduction from the minimax equilibrium to the DBR problem follows a standard primal-dual argument. What is interesting, however, is the other direction – i.e., solving the general 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 objective over $\mathcal{E}$. Moreover, security games are very well-structured 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. There has been a belief in the literature on the possibility of 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 [5], various techniques from convex and non-convex optimization as well as heuristics. However, the message conveyed by Theorem 4.1 is that to solve security games, one cannot avoid solving, and also only needs to solve, the DBR problem, if polynomial time efficiency is concerned.

To prove that the DBR problem reduces to equilibrium computation, our reduction makes use of the polynomial time equivalence between optimization and membership checking for a polytope $\mathcal{P}$. To test whether any given $\mathbf{x}$ is in $\mathcal{P}$ or not, the natural idea is to construct a zero-sum security game instance in which the defender's maximin strategy corresponds to $\mathbf{x}$ precisely when $\mathbf{x} \in \mathcal{P}$. However, unfortunately, this is not always possible, because some $\mathbf{x}$ is entry-wise dominated by other $\mathbf{x}' \in \mathcal{P}$, thus $\mathbf{x}$ can never be a maximin strategy. To overcome this barrier, we instead consider membership checking and optimization over a "relaxed" polytope $\widehat{\mathcal{P}}$. We show that if there is a zero-sum security game with defender maximin strategy $\mathbf{x}^* \in \mathcal{P}$, then any $\mathbf{x}$ that is dominated by $\mathbf{x}^*$ will be in $\widehat{P}$. Such a $\widehat{\mathcal{P}}$ satisfies the *down-monotone* property. This resolves the difficulty of checking some $\mathbf{x}$ that can never be a maximin strategy. Given the ability of checking membership for $\widehat{\mathcal{P}}$, we can optimize over $\widehat{\mathcal{P}}$. Finally, we make use of the condition $\mathbf{w} \geq 0$ (essentially due to the assumption $r_i > c_i$) to show how to transfer an optimization problem over $\widehat{\mathcal{P}}$ back to optimization over $\mathcal{P}$. Due to space considerations, we omit details here and refer the reader to the full version for a formal proof.

## 4.1 When Gaming is Easier than Best Response

Security games are well-structured 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 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. We answer this in the affirmative for security games. Notice that our construction does not contradict Theorem 4.1 since the security game instances we will construct have further constrained payoffs, which make solving the game easy but still maintain the hardness of the best response.

PROPOSITION 4.2. *There exist zero-sum security games such that the minimax equilibrium can be computed in polynomial time but the DBR problem is NP-hard.*

PROOF. Consider a security game played on a complete graph $K_n$. Each *edge* is a target. The defender has $k(< n)$ security resources and each resource can patrol a *vertex*, by which the $n-1$ adjacent edges of this vertex are covered. An edge is covered if at least one of its end vertexes is patrolled. Given that the attacker attacks any edge $e$, the defender gets utility $1$ if it is protected and utility $0$ otherwise. The attacker seeks to minimize the defender's utility. The DBR problem for this security game is, given weight $w_e \geq 0$ for any edge $e \in K_n$, finding $k$ vertexes that maximize the total edge weights they cover. This is NP-hard by a trivial reduction from vertex cover.

We now show that the minimax equilibrium of this game can be computed in $poly(n)$ time. In fact, we claim that uniformly randomly sampling $k$ vertexes from $K_n$ to patrol is a minimax equilibrium. Observe that any edge in the constructed mixed strategy is covered with equal probability $\frac{k}{n} + \frac{k}{n} - \frac{k(k-1)}{n(n-1)}$ where the last term is the probability that both end vertexes of an edge is patrolled. Moreover, any pure strategy, represented as a binary vector, has precisely $k(n-1) - k(k-1)/2$ entries of value 1. This is because any $k$ vertexes in $K_n$ covers those many edges. As a result, for the marginals $\mathbf{x}$ of any mixed strategy, the sum of its entry values is also $k(n-1) - k(k-1)/2$. Since each edge has the same value to the defender, the optimal way is to distribute these probability mass evenly to the $n(n-1)/2$ edges and each edge gets probability mass

$$\frac{k(n-1) - k(k-1)/2}{n(n-1)/2} = \frac{2k}{n} - \frac{k(k-1)}{n(n-1)}.$$

However, this is precisely what the constructed mixed strategy is achieving. Therefore, sampling $k$ vertexes uniformly at random is a minimax equilibrium. □

# 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 $\rho_i$ [$\zeta_i$] is the attacker's reward [cost], if target $i$ is attacked. We consider the computation of the two mostly adopted equilibrium concepts in 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 $poly(n)$ time algorithm to compute the Strong Stackelberg equilibrium for security games over $\mathcal{E}$,* if and only if *there is a $poly(n)$ 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. Recall the definition of SSE in Section 2, without loss of generality, we can assume the attacker always plays a pure strategy since he moves after the defender. Therefore, as observed in [7], to compute the defender's optimal mixed strategy, we only need to enumerate all the possibilities of the attacker's best response choices. In particular, constrained on that the attacker's best response is target $k$, the defender's optimal strategy can be computed by the following linear program (denoted as $LP_k$):

$$\begin{array}{ll} \text{maximize} & x_k r_k + (1-x_k)c_k \\ \text{subject to} & (1-x_k)\rho_k + x_k\zeta_k \geq \\ & \quad (1-x_i)\rho_i + x_i\zeta_i, \quad \text{for } i \neq k. \\ & \sum_{\mathbf{e}\in\mathcal{E}} p_e \cdot \mathbf{e} = \mathbf{x} \\ & \sum_{\mathbf{e}\in\mathcal{E}} p_e = 1 \\ & p_e \geq 0, \qquad\qquad \text{for } \mathbf{e} \in \mathcal{E}. \end{array}$$

where the first constrain is to guarantee that the attacker is indeed incentivized to attack target $k$. Similar to the reduction in Theorem 4.1, we can show that $LP_k$ can be solved in $poly(n)$ time with access to a $poly(n)$ time DBR oracle by exploring the dual program. The SSE can be computed by solving $LP_1, ..., LP_n$ and then picking the defender mixed strategy (and corresponding attacker best response) from the LP of the largest objective. □

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 [9, 6], and is only harder for general bilinear games [14]. Interestingly, it turns out that computing a 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. [20].

LEMMA 5.2. *(Korzhyk et al. [20]) Consider a security game $\mathcal{G}(\mathbf{r}, \mathbf{c}, \rho, \zeta, \mathcal{E})$. Let $\overline{\mathcal{G}}(-\zeta, -\rho, \rho, \zeta, \mathcal{E})$ 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 $\overline{\mathcal{G}}$, where the one-to-one transform function $f : \mathbb{R}^n \to \mathbb{R}^n$ is defined as follows:*

$$f_i(\mathbf{y}) = \frac{1}{\lambda} \frac{r_i - c_i}{\rho_i - \zeta_i} y_i, \ \forall i \in [n], \tag{4}$$

*where $\lambda = \sum_{i=1}^{n} \frac{r_i - c_i}{\rho_i - \zeta_i} y_i$ is the normalization factor. 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 receives the same utility in any Nash equilibrium of $\mathcal{G}$.*

Notice that the transform function defined in Equation (4) is *non-linear* due to the normalization factor. We provide an intuitive explanation of Lemma 5.2 here, while refer the reader to [20] 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 $\overline{\mathcal{G}}$, 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 $\overline{\mathcal{G}}$, 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 $\overline{\mathcal{G}}$. It is easy to see that the attacker's utility in any NE equals his (unique) utility in the zero-sum game $\overline{\mathcal{G}}$.

As a corollary of Lemma 5.2 and Theorem 4.1, *one* Nash equilibrium of a security game can be computed in polynomial time if and only if the DBR problem over $\mathcal{E}$ admits a polynomial time algorithm. However, it is widely known that the Nash equilibrium is not unique in games. To predict and analyze the game outcomes, we usually need to compute some particular Nash equilibria, among which the NE that maximizes or minimizes the defender's utility are undoubtedly the most desirable ones. 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 [20]).

As widely known, maximizing a player's utility over Nash equilibria is NP-hard even in two-player normal-form games [15, 8]. It will be appealing if the optimal NE can be efficiently computed in the security game, which is widely recognized as a very successful application of game theory. Our next result shows that this is indeed the case! The technical barrier here is, the defender's equilibrium utility is a convex function of the attacker's mixed strategy (as we will show), and it is generally hard to maximize a convex function. Interestingly, we prove that the defender equilibrium utility becomes linear when restricted to the domain of the attacker *equilibrium strategies*. Due to space limit, we omit the proof here and refer the reader to the full version.

THEOREM 5.3. *There is a $poly(n)$ 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 $poly(n)$ 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.*

# 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 captured by the set system $\mathcal{E}$. Therefor, it should only be a modeling choice, while not the computational concern, about which equilibrium concept to pick in real applications.

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 Strong Stackelberg equilibrium for security games over $\mathcal{E}$;*
*(4) Computing best/worst (for the defender) Nash equilibrium for security games over $\mathcal{E}$.*

Corollary 6.1 provides a more convenient way for us 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 flow formulation, the equilibrium of the security game can be computed efficiently. On the other hand, when the optimization over $\mathcal{E}$ is a coverage problem, even vertex coverage or 2-D geometric coverage (NP-hardness proved in [12]), a packing problem or a Knapsack problem, the equilibrium computation for these security games are NP-hard in general.

Finally, our framework provides a more combinatorial way to think about security games. It can also be used to easily recover and strengthen some known complexity results in the literature of security games, as well as to resolve some open problems from former work. For example, Xu et al. [35] considered the computation of the minimax equilibrium in zero-sum spatio-temporal security games (see Section 3.3). They proved that the DBR problem there is NP-hard in general, but the computation of minimax equilibrium is left open. Theorem 4.1 resolves this open question. The NP-hardness of the Transportation Security Administration (TSA) problem shown in [4] easily follows from Corollary 6.1

and the fact that independent set is NP-hard. In [13], Gan et al. consider security games on graphs where targets are vertexes. The defender chooses a subset of vertexes to patrol, by which the patrolled vertexes as well as their adjacent vertexes are covered. The DBR problem is to, given a non-negative weight for each vertex, find a subset of vertexes to cover so that the total weight of the covered vertexes is maximized. This is NP-hard by a simple reduction from set cover. Gan et al. show that computing the SSE is NP-hard. Our results indicate that computing the minimax equilibrium is also NP-hard. Korzhyk et al. [18] consider security games in which the DBR problem is a the coverage problem as discussed in Section 3.3. They show 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 [22] proved complexity results for security games played on graphs where vertexes are targets and each resource can patrol an edge or path of the graph. Some results can be easily recovered under our framework as well. For example, the DBR problem corresponding to the two positive results there can be solved respectively by a greedy algorithm (the case of Theorem 1 of [22]) and dynamic programming (the case of Theorem 2 in [22]). We omit further details here.

# 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 an important direction 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 transfer an FPTAS for the DBR problem to an algorithm for computing an $\epsilon$-minimax equilibrium (see [17]), but the reverse direction and other generalizations are open. Second, the results of computing the best/worst Nash equilibrium is surprising. We wonder to what extent similar results can hold, and in particular, whether there are other interpretable game classes where this is true. Finally, there are several ways to generalize our model. For example, the players' utility functions may not be linear, but can still by compactly represented and computed. One particular example is the network interdiction game played on a graph [34, 32], 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 [19]. We wonder how the computational complexity of the proposed four problems relates to each other in these generalized settings.


## Acknowledgement

We thank Shaddin Dughmi, Milind Tambe and Vincent Conitzer for helpful discussions.


# REFERENCES


[1] B. An, E. Shieh, M. Tambe, R. Yang, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. PROTECT–a deployed game theoretic system for strategic security allocation for the United States Coast Guard. *AI Magazine*, 33(4):96, 2012.

[2] S. Bhattacharya, V. Conitzer, and K. Munagala. Approximation algorithm for security games with costly resources. In *Internet and Network Economics*, pages 13–24. Springer, 2011.

[3] V. Bier, S. Oliveros, and L. Samuelson. Choosing what to protect: Strategic defensive allocation against an unknown attacker. *Journal of Public Economic Theory*, 9(4):563–587, 2007.

[4] M. Brown, A. Sinha, A. Schlenker, and M. Tambe. One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In *AAAI conference on Artificial Intelligence (AAAI)*, 2016.

[5] E. Budish, Y.-K. Che, F. Kojima, and P. Milgrom. Designing random allocation mechanisms: Theory and applications. *The American Economic Review*, 103(2):585–623, 2013.

[6] X. Chen, X. Deng, and S.-H. Teng. Settling the complexity of computing two-player Nash equilibria. *Journal of the ACM (JACM)*, 56(3):14, 2009.

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

[8] V. Conitzer and T. Sandholm. New complexity results about Nash equilibria. *Games and Economic Behavior*, 63(2):621–641, 2008.

[9] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a Nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

[10] F. Fang, A. X. Jiang, and M. Tambe. Optimal patrol strategy for protecting moving targets with multiple mobile resources. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems*, pages 957–964. International Foundation for Autonomous Agents and Multiagent Systems, 2013.

[11] F. Fang, T. H. Nguyen, R. Pickles, W. Y. Lam, G. R. Clements, B. An, A. Singh, M. Tambe, and A. Lemieux. Deploying PAWS: Field optimization of the protection assistant for wildlife security. In *Proceedings of the Twenty-Eighth Innovative Applications of Artificial Intelligence Conference*, 2016.

[12] R. J. Fowler, M. S. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are NP-complete. *Information processing letters*, 12(3):133–137, 1981.

[13] J. Gan, B. An, and Y. Vorobeychik. Security games with protection externalities. In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[14] J. Garg, A. X. Jiang, and R. Mehta. Bilinear games: Polynomial time algorithms for rank based subclasses. In *Internet and Network Economics*, pages 399–407. Springer, 2011.

[15] I. Gilboa and E. Zemel. Nash and correlated equilibria: Some complexity considerations. *Games and Economic Behavior*, 1(1):80–93, 1989.

[16] M. Grötschel, L. Lovász, and A. Schrijver. *Geometric algorithms and combinatorial optimization*, volume 2. Springer Science & Business Media, 2012.

[17] N. Immorlica, A. T. Kalai, B. Lucier, A. Moitra, A. Postlewaite, and M. Tennenholtz. Dueling algorithms. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 215–224. ACM, 2011.

[18] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*, 2010.

[19] D. Korzhyk, V. Conitzer, and R. Parr. Security games with multiple attacker resources. In *IJCAI*, 2011.

[20] D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, and M. Tambe. Stackelberg vs. Nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness. *J. Artif. Int. Res.*, 41(2):297–327, May 2011.

[21] V. E. Leeuwen et al. *Optimization and approximation on systems of geometric objects*. 2009.

[22] J. Letchford and V. Conitzer. Solving security games on graphs via marginal probabilities. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*, 2013.

[23] J. A. Major. Advanced techniques for modeling terrorism risk. *The Journal of Risk Finance*, 4(1):15–24, 2002.

[24] M. Mavronicolas, L. Michael, V. Papadopoulou, A. Philippou, and P. Spirakis. The price of defense. In *Mathematical Foundations of Computer Science 2006*, pages 717–728. Springer, 2006.

[25] M. Mavronicolas, V. Papadopoulou, A. Philippou, and P. Spirakis. A graph-theoretic network security game. In *Internet and Network Economics*, pages 969–978. Springer, 2005.

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

[27] T. Sandler et al. Terrorism & game theory. *Simulation & Gaming*, 34(3):319–337, 2003.

[28] T. Sandler et al. Counterterrorism a game-theoretic analysis. *Journal of conflict resolution*, 49(2):183–200, 2005.

[29] A. Schrijver. *Theory of linear and integer programming*. John Wiley & Sons, 1998.

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

[31] J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. IRIS - a tool for strategic security allocation in transportation networks. In *The Eighth International Conference on Autonomous Agents and Multiagent Systems - Industry Track*, 2009.

[32] J. Tsai, Z. Yin, J.-y. Kwak, D. Kempe, C. Kiekintveld, and M. Tambe. Urban security: Game-theoretic resource allocation in networked physical domains. In *AAAI*, 2010.

[33] Y. Vorobeychik and S. P. Singh. Computing Stackelberg equilibria in discounted stochastic games. In *AAAI*, 2012.

[34] A. Washburn and K. Wood. Two-person zero-sum games for network interdiction. *Operations Research*, 43(2):243–251, 1995.

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

[36] R. Yang, B. Ford, M. Tambe, and A. Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 453–460. International Foundation for Autonomous Agents and Multiagent Systems, 2014.

[37] Z. Yin, A. X. Jiang, M. Tambe, C. Kiekintveld, K. Leyton-Brown, T. Sandholm, and J. P. Sullivan. TRUSTS: Scheduling randomized patrols for fare inspection in transit systems using game theory. *AI Magazine*, 33(4):59, 2012.