PROTECT in the Ports of Boston, New York and Beyond: Experiences in Deploying Stackelberg Security Games with Quantal Response

Eric Shieh, Bo An, Rong Yang, Milind Tambe, Craig Baldwin, Joseph DiRenzo, Ben Maule, Garrett Meyer, Kathryn Moretti

Abstract While three deployed applications of game theory for security have recently been reported at AAMAS [21], we as a community remain in the early stages of these deployments; there is a continuing need to understand the core principles for innovative security applications of game theory. Towards that end, this chapter presents PROTECT, a game-theoretic system deployed by the United States Coast Guard (USCG) in the port of Boston for scheduling their patrols. USCG has termed the deployment of PROTECT in Boston a success, and efforts are underway to test it in the port of New York, with the potential for nationwide deployment.
PROTECT is premised on an attacker-defender Stackelberg game model and offers five key innovations. First, this system is a departure from the assumption of perfect adversary rationality noted in previous work, relying instead on a quantal response (QR) model of the adversary's behavior. To the best of our knowledge, this is the first real-world deployment of the QR model. Second, to improve PROTECT's efficiency, we generate a compact representation of the defender's strategy space, exploiting equivalence and dominance. Third, we show how to practically model a real maritime patrolling problem as a Stackelberg game. Fourth, our experimental results illustrate that PROTECT's QR model more robustly handles real-world uncertainties than a perfect rationality model does. Finally, in evaluating PROTECT, this chapter provides real-world data: (i) comparison of human-generated vs. PROTECT security schedules, and (ii) results from an Adversarial Perspective Team's (human mock attackers) analysis.

Eric Shieh, Bo An, Rong Yang, Matthew P. Johnson, Milind Tambe
University of Southern California, {eshieh, boa, yangrong, tambe}@usc.edu

Craig Baldwin, Joseph DiRenzo, Ben Maule, Garrett Meyer, Kathryn Moretti
United States Coast Guard, {craig.w.baldwin, joseph.direnzo, ben.j.maule, garrett.r.meyer, kathryn.a.moretti}@uscg.mil

1 Introduction

The global need for security of key infrastructure with limited resources has led to significant interest in research conducted in multiagent systems towards game-theory for real-world security. As reported previously at AAMAS, three applications based on Stackelberg games have been transitioned to real-world deployment. This includes ARMOR, used by the Los Angeles International Airport to randomize checkpoints of roadways and canine patrols [16]; IRIS, which helps the US Federal Air Marshal Service [22] in scheduling air marshals on international flights; and GUARDS [17], which is under evaluation by the US Transportation Security Administration to allocate resources for airport protection. We as a community remain in the early stages of these deployments, and must continue to develop our understanding of core principles of innovative applications of game theory for security.

To this end, this chapter presents a new game-theoretic security application to aid the United States Coast Guard (USCG), called *Port Resilience Operational/Tactical Enforcement to Combat Terrorism* (PROTECT), which originally appeared [20]. The USCG's mission includes maritime security of the US coasts, ports, and inland waterways; a security domain that faces increased risks in the context of threats such as terrorism and drug trafficking. Given a particular port and the variety of critical infrastructure that an adversary may attack within the port, the USCG conducts patrols to protect this infrastructure; however, while the adversary has the opportunity to observe patrol patterns, limited security resources mean that USCG patrols cannot be at every location 24/7. To assist the USCG in allocating its patrolling resources, similar to previous applications [16, 17, 22], PROTECT uses an attacker-defender Stackelberg game framework, with USCG as the defender against terrorist adversaries that conduct surveillance before potentially launching an attack. PROTECT's solution is to typically provide a mixed strategy, i.e. randomized patrol patterns taking into account the importance of different targets, and the adversary's surveillance and anticipated reaction to USCG patrols.

While PROTECT builds on previous work, this paper highlights five key innovations. The first and most important is PROTECT's departure from the assumption of perfect rationality on the part of the human adversaries. While appropriate in the initial applications (ARMOR, IRIS, GUARDS) this assumption of perfect rationality is well-recognized as a limitation of classical game theory, and bounded rationality has received significant attention in behavioral game-theoretic approaches [4]. Within this behavioral framework, quantal response equilibrium has emerged as a promising approach to model human bounded rationality [4, 14, 24] including recent results illustrating the benefits of the quantal response (QR) model in security games contexts [25]. Therefore, PROTECT uses a novel algorithm called PASAQ [26] based on the QR model of a human adversary. To the best of our knowledge, this is the *first time that the QR model has been used in a real-world security application*.

Second, PROTECT improves PASAQ's efficiency via a compact representation of defender strategies exploiting dominance and equivalence analysis. Experimental results show the significant benefits of this compact representation. Third, PROTECT addresses practical concerns of modeling a real-world maritime patrolling

application in a Stackelberg framework. Fourth, this chapter presents a detailed simulation analysis of PROTECT's robustness to uncertainty that may arise in the real-world. For various cases of added uncertainty, the chapter shows that PROTECT's quantal-response-based approach leads to significantly improved robustness when compared to an approach that assumes full attacker rationality.

PROTECT has been in use at the port of Boston since April 2011 and been evaluated by the USCG. This evaluation brings forth our final key contribution: for the first time, this paper provides real-world data comparing human-generated and game-theoretic schedules. We also provide results from an Adversarial Perspective Team's (APT) analysis and comparison of patrols before and after the use of the PROTECT system from a viewpoint of an attacker. Given the success of PROTECT in Boston, we are now extending it to the port of New York, which is a much larger, more complex security environment. Based on the outcome there, it may potentially be extended to other ports in the US.

(a) PROTECT is being used in Boston.

(b) Extending PROTECT to New York.

Fig. 1 USCG boats patrolling the ports of Boston and New York.

2 Background

In this section, we explain the framework for Stackelberg security games, for which PROTECT is modeled. Previous deployed security applications that are based on Stackelberg games are then presented.

2.1 Stackelberg Security Game

Recent security games have increasing relied on a Stackelberg framework as Stackelberg games are based on a model of a leader and follower. In a security domain, the defender has a set of targets that must be defended with only a limited number

of resources. The attacker is able to observe the strategy of the defender and utilize this information in planning an attack. This fits the description of a Stackelberg game with the defender in the role of the leader and the attacker in the role of the follower. In the security domain, an action, or *pure strategy*, for the defender is an allocation of resources on a set of targets or patrols, e.g., list of checkpoints at the LAX airport or flight schedule for federal air marshals. The pure strategy of an attacker is a target, e.g., an airport terminal or a flight. The strategy for the leader, or defender, is a *mixed strategy*, i.e., a probability distribution over the pure strategies. The strategy for the follower, or attacker, is also a mixed strategy. However, dependent on the model of the attacker, the mixed strategy for the attacker may be equivalent to a pure strategy where a particular pure strategy, or target, is selected with a probability of 1 while all other strategies have a probability of 0. For each target, there exists payoff values that represents the utility for both the attacker and defender in the scenario of either a successful or failed attack.

In a security game, each target has a set of four payoff values: the reward and penalty for both the defender and attacker in the case of a successful or failed attack. These four payoff values are enough for all possible outcomes in the security domain. Table 1 gives an example of a Stackelberg security game composed of two targets, t_1 and t_2, and the four corresponding payoffs for each one. In this example, if the defender chose to cover or protect target t_1 and the attacker attacked target t_1, then the defender would receive a reward of 5 while the attacker would receive a penalty of -4.

	Defender		Attacker	
Target	Covered	Uncovered	Covered	Uncovered
t_1	5	-3	-4	4
t_2	10	-7	-1	2

Table 1 Payoff table for example Stackelberg security game.

An assumption made by security games is that the defender's payoff for covering a target is greater than the payoff for leaving a target uncovered, and that the attacker's payoff for attacking an uncovered target is greater than the payoff for attacking a target that is uncovered. This assumption holds in the real-world. Another aspect of security games is that the payoff of a scenario is dependent only on the target that is attacked and whether or not it is covered by the defender [10]. The payoff does not rely on the defender coverages on the targets that are not attacked. For example, if the attacker decides to attack target t_1, whether or not the defender covers target t_2 does not impact the final payoff value. This aids in computing the defender's optimal strategy because many resource allocations of the defender are identical.

2.2 Deployed Security Applications

The Stackelberg security game framework has been used and in the following deployed security applications: ARMOR [16], IRIS [22], and GUARDS [17]. The ARMOR system (Assistant for Randomized Monitoring over Routes) has been at use in the Los Angeles International Airport (LAX) since 2007 in scheduling checkpoints on inbound roads into the airport and patrol routes for bomb-sniffing canine units in the eight terminals of LAX. The terminals in LAX have various characteristics such as passenger population, physical size, and international or domestic flights. This impacts the risk assessment and subsequently the payoffs for each individual terminal. The LAX personnel are limited in both the human resources available to conduct checkpoints and the number of canine units. ARMOR is used to optimally allocate the resources in such a way as to improve effectiveness of the checkpoints and canine patrols while also avoiding patterns that an adversary may exploit. It uses a Bayesian Stackelberg solver known as DOBSS (Decomposed Optimal Bayesian Stackelberg Solver) [15] to compute the optimal mixed strategy for both where to set up checkpoints and the terminals for the canine units.

A second application is called IRIS (Intelligent Randomization In Scheduling) and has been used by the US Federal Air Marshals Service (FAMS) since 2009 in scheduling air marshals to flights that originate in and depart from the United States. The purpose of the air marshals is to both dissuade potential adversaries and prevent possible attacks. Flights have different values based on various factors such as the number of passengers, the source/destination countries and cities, and special events that can impact the risk for certain flights. IRIS is used to schedule tours of flights that not only obey different constraints (e.g., boarding time or flight time), but also must take into account the limited number of air marshals and varying values of each flight. Due to the exponential increase in the possible number of schedules based on the number of flights and resources, DOBSS was not applicable and thus the faster ASPEN [8] solver was used to generate the thousands of flights per day.

The third security application that is based on the Stackelberg security game framework is GUARDS (Game-theoretic Unpredictable and Randomly Deployed Security). GUARDS is used by the United States Transportation Security Administration (TSA) in scheduling resources to conduct various security activities. While many people are aware of common security activities, such as passenger screening, this is just one of many security layers TSA personnel implement to help prevent potential threats. These layers can involve hundreds of heterogeneous security activities executed by limited TSA personnel leading to a complex resource allocation challenge where the TSA is unable to run every security activity all the time [17]. To address the many heterogeneous security activities, GUARDS created a new game-theoretic framework that takes into account heterogeneous defender activities along with a compact modeling of the large number of threats.

3 USCG and PROTECT's Goals

The USCG continues to face challenges with evolving asymmetric threats within the maritime environment not only within the maritime global commons, but also within the ports and waterways that make up the United States Maritime Transportation System. The former Director of National Intelligence, Dennis Blair noted in 2010 a persistent threat "from al-Qa'ida and potentially others who share its anti-Western ideology. A major terrorist attack may emanate from either outside or inside the United States" [3]. This threat was reinforced in May of 2011 following the raid on Osama Bin Laden's home, where a large trove of material was uncovered, including plans to attack an oil tanker. "There is an indication of intent, with operatives seeking the size and construction of tankers, and concluding it's best to blow them up from the inside because of the strength of their hulls" [6]. These oil tankers transit the U.S. Maritime Transportation System. The USCG plays a key role in the security of this system and the protection of seaports to support the economy, environment, and way of life in the US.

Coupled with challenging economic times, USCG must operate as effectively as possible, achieving maximum benefit from every hour spent on patrol. As a result, USCG is compelled to re-examine the role that optimization of security resource usage plays in its mission planning—and how innovations provided by game theory can be effectively employed.

The goal of PROTECT is to use game theory to assist the USCG in maximizing its effectiveness in its Ports, Waterways, and Coastal Security (PWCS) Mission. PWCS patrols are focused on protecting critical infrastructure; without the resources to provide one hundred percent on-scene presence at any, let alone all, parts of the critical infrastructure, optimization of security resource is critical. Towards that end, unpredictability creates situations of uncertainty for an enemy and can be enough to deem a target less appealing.

The PROTECT system, focused on the PWCS patrols, addresses how the USCG should optimally patrol critical infrastructure in a port to maximize protection, knowing that the adversary may conduct surveillance and then launch an attack. While randomizing patrol patterns is key, PROTECT also addresses the fact that the targets are of unequal value, understanding that the adversary will adapt to whatever patrol patterns USCG conducts. The output of PROTECT is a schedule of patrols which includes when the patrols are to begin, what critical infrastructure to visit for each patrol, and what activities to perform at each critical infrastructure. While initially pilot-tested in the port of Boston, the solution technique was intended to be generalizable and applicable to other ports.

4 Key Innovations in PROTECT

The PWCS patrol problem was modeled as a leader-follower (or attacker-defender) Stackelberg game [7] with USCG as the leader (defender) and the terrorist adver-

saries in the role of the follower (attacker). The choice of this framework was supported by prior successful applications of Stackelberg games [21]. In this Stackelberg game framework, the defender commits to a mixed (randomized) strategy of patrols, whereas the attacker conducts surveillance of these mixed strategies and responds with a pure strategy of an attack on a target. The objective of this framework is to find the optimal mixed strategy for the defender.

Stackelberg games are well established in the multi-agent systems literature [5, 11, 13, 21]. The last several years have witnessed the successful application of multi-agent systems in allocating limited resources to protect critical infrastructures [2, 12, 9, 17]. The framework of game-theory, and more specifically, of Stackelberg games, is well suited to formulate the strategic interaction in security domains in which there are usually two players: the security force (defender) commits to a security policy first and the attacker (e.g., terrorist, poacher and smuggler) conducts surveillance to learn the policy and then takes his best attacking action.[1] Stackelberg games have been widely used for modeling/reasoning complex security problems and a variety of algorithms have been proposed to efficiently compute the equilibrium strategy, i.e., the defender's best way of utilizing her limited security resources. (There is actually a special class of Stackelberg games that often gets used in these security domains, and this class is referred to as security games.) In the rest of this section, we describe the application of the Stackelberg game framework in multiple significant security domains.

We will now discuss three of PROTECT's key innovations over previous such works. We begin by discussing how to practically cast this real-world maritime patrolling problem of PWCS patrols as a Stackelberg game (Section 4.1). We also show how to reduce the number of defender strategies (Section 4.2) before addressing the most important of the innovations in PROTECT: its use of the quantal response model (Section 4.3).

4.1 Game Modeling

To model the USCG patrolling domain as a Stackelberg game, we need to define (i) the set of attacker strategies, (ii) the set of defender strategies, and (iii) the payoff function. These strategies and payoffs center on the targets in a port, for example pieces of critical infrastructure. In our Stackelberg game formulation, the attacker conducts surveillance on the mixed strategies that the defender has committed to, and can then launch an attack. Thus, the attacks an attacker can launch on different possible targets are considered as his/her pure strategies.

However, the definition of defender strategies is not as straightforward. Patrols last for some fixed duration during the day as specified by USCG, e.g. 4 hours. Our first attempt was to model each target as a node in a graph and allow patrol paths to go from each individual target to (almost all) other targets in the port, generating

[1] Or the attacker may be sufficiently deterred and dissuaded from attacking the protected target.

Patrol Schedule	Target 1	Target 2	Target 3	Target 4
(1:k_1), (2:k_1), (1:k_1)	50,-50	30,-30	15,-15	-20,20
(1:k_2), (2:k_1), (1:k_1)	100,-100	60,-60	15,-15	-20,20
(1:k_1), (2:k_1), (1:k_2)	100,-100	60,-60	15,-15	-20,20
(1:k_2), (2:k_1), (1:k_2)	100,-100	60,-60	15,-15	-20,20
(1:k_1), (3:k_1), (2:k_1), (1:k_1)	50,-50	30,-30	15,-15	10,-10
(1:k_1), (2:k_1), (3:k_1), (1:k_1)	50,-50	30,-30	15,-15	10,-10

Table 2 Portion of a simplified example of a game matrix

an almost complete graph on the targets. This method yields the most flexible set of patrol routes that would fit within the maximum duration, covering any permutation of targets within a single patrol. This method unfortunately faced significant challenges: (i) it required determining the travel time for a patrol boat for each pair of targets, a daunting knowledge acquisition task given the hundreds of pairs of targets; (ii) it did not maximize the use of port geography whereby boat crews could observe multiple targets at once and; (iii) it was perceived as micromanaging the activities of the USCG boat crews, which was undesirable.

Our improved approach to generating defender strategies therefore grouped nearby targets into patrol areas. The presence of patrol areas led the USCG to redefine the set of defensive activities to be performed on patrol areas to provide a more accurate and expressive model of the patrols. Activities that take a longer time provide the defender a higher payoff compared to activities that take a shorter time to complete. This impacts the final patrol schedule as one patrol may visit fewer areas but conduct longer-duration defensive activities at the areas, while another patrol may have more areas with shorter-duration activities.

To generate all the permutations of patrol schedules, a graph $\mathscr{G} = (\mathscr{V}, \mathscr{E})$ is created with the patrol areas as vertices $\mathscr{V}$ and adjacent patrol areas as edges $\mathscr{E}$. Using the graph of patrol areas, PROTECT generates all possible patrol schedules, each of which is a closed walk of $\mathscr{G}$ that starts and ends at the patrol area $b \in \mathscr{V}$, the base patrol area for the USCG. The patrol schedules are a sequence of patrol areas and associated defensive activities, and are constrained by a maximum patrol time τ.

The graph $\mathscr{G}$ along with the constraints b and τ are used to generate the defender strategies (patrol schedules). Given each patrol schedule, the total patrol schedule time is calculated (this also includes traversal time between areas, but we ignore this for expository purposes); we then verify that the total time is less than or equal to the maximum patrol time τ. After generating all possible patrol schedules, a game is formed where the set of defender strategies is composed of patrol schedules and the set of attacker strategies is the set of targets. The attacker's strategy was based on targets instead of patrol areas because an attacker will choose to attack a single target.

Table 2 gives an example, where the rows correspond to the defender's strategies and the columns correspond to the attacker's strategies. In this example, there are two possible defensive activities, activity k_1 and k_2, where k_2 provides a higher payoff for the defender than k_1. Suppose that the time bound disallows more than two k_2 activities (given the time required for k_2) within a patrol. Patrol area 1 has

two targets (numbered 1 and 2) while patrol areas 2 and 3 each have one target (numbered 3 and 4, respectively). In the table, a patrol schedule is composed of a sequence of patrol areas and a defensive activity in each area. The patrol schedules are ordered so that the first patrol area in the schedule denotes which patrol area the defender needs to visit first. In this example, patrol area 1 is the base patrol area, and all of the patrol schedules begin and end at patrol area 1. For example, the patrol schedule in row 2 first visits patrol area 1 with activity k_2, then travels to patrol area 2 with activity k_1, and finally returns to patrol area 1 with activity k_1. For the payoffs, if a target i is the attacker's choice and is also part of a patrol schedule, then the defender would gain a reward R_i^d while the attacker would receive a penalty P_i^a, else the defender would receive a penalty P_i^d and the attacker would gain a reward R_i^a. Furthermore, let G_{ij}^d be the payoff for the defender if the defender chooses patrol j and the attacker chooses to attack target i. G_{ij}^d can be represented as a linear combination of the defender reward/penalty on target i and A_{ij}, the effectiveness probability of the defensive activity performed on target i for patrol j, as described by Equation 1. The value of A_{ij} is 0 if target i is not in patrol j.

$$G_{ij}^d = A_{ij}R_i^d + (1 - A_{ij})P_i^d \tag{1}$$

For instance, suppose target 1 is covered using k_1 in strategy 5, and the value of A_{15} is 0.5. If $R_1^d = 150$ and $P_1^d = -50$, then $G_{15}^d = 0.5(150) + (1-0.5)(-50) = 50$. ($G_{ij}^a$ would be computed in a similar fashion.) If a target is visited multiple times with different activities, only the highest quality activity is considered.

In the USCG problem, rewards and penalties are based on an analysis completed by a contracted company of risk analysts that looked at the targets in the port of Boston and assigned corresponding values for each one. The types of factors taken into consideration for generating these values include economic damage and injury/loss of life. Meanwhile, the effectiveness probability values A_{ij} for different defensive activities are decided based on the duration of the activities. Longer activities lead to a higher probability of capturing the attackers. While Table 2 shows a zero-sum game, the algorithm used by PROTECT is *not limited to a zero-sum game*; the actual payoff values are determined by the USCG.

4.2 Compact Representation

In our game, the number of defender strategies, i.e. patrol schedules, grows combinatorially, generating a scale-up challenge. To achieve scale-up, PROTECT uses a compact representation of the patrol schedules using two ideas: (i) combining equivalent patrol schedules, and (ii) removal of dominated patrol schedules.

With respect to equivalence, different permutations of patrol schedules provide identical payoff results. Furthermore, if an area is visited multiple times with different activities in a schedule, only the activity that provides the defender the highest payoff requires attention. Therefore, many patrol schedules are equivalent if the set

of patrol areas visited and defensive activities in the schedules are the same, even if their order differs. Such equivalent patrol schedules are combined into a single compact defender strategy, represented as a set of patrol areas and defensive activities (and omitting any ordering information). Table 3 presents a compact version of Table 2, with the game matrix simplified using equivalence. For example, the patrol schedules in the rows 2-4 from Table 2 are represented as a compact strategy Γ_2 = $\{(1,k_2), (2,k_1)\}$ in Table 3.

Compact Strategy	Target 1	Target 2	Target 3	Target 4
$\Gamma_1 = \{(1{:}k_1), (2{:}k_1)\}$	50,-50	30,-30	15,-15	-20,20
$\Gamma_2 = \{(1{:}k_2), (2{:}k_1)\}$	100,-100	60,-60	15,-15	-20,20
$\Gamma_3 = \{(1{:}k_1), (2{:}k_1), (3{:}k_1)\}$	50,-50	30,-30	15,-15	10,-10

Table 3 Example compact strategies and game matrix.

Next, the idea of dominance is illustrated in Table 3. Note the difference between Γ_1 and Γ_2. Since activity k_2 gives the defender a higher payoff than k_1, Γ_1 can be removed from the set of defender strategies because Γ_2 covers the same patrol areas while giving a higher payoff for patrol area 1. To generate the set of compact defender strategies, a naive approach would be to first generate the full set of patrol schedules and then prune the dominated and equivalent schedules. Instead, PROTECT uses three ideas to quickly compute the compact strategies: (i) computation of a starting point for compact strategy generation; (ii) computation of a stopping point and; (iii) verification of feasibility in compact strategies.

While generating compact strategies, we first generate compact strategies containing $\hat{n}$ patrol areas, then $\hat{n}-1$ patrol areas and so on down to $\check{n}$ patrol areas. $\hat{n}$ is called the starting point and is defined as τ/ρ where τ is the maximum patrol time and ρ shortest duration of a defensive activity. The maximum number of areas in any compact strategy must be less than or equal to $\hat{n}$. For example, if there are 20 patrol areas, τ =100 minutes and ρ = 10 minutes, then the algorithm will start by generating compact strategies with 10 patrol areas. It must be verified that a feasible patrol schedule can be formed from each compact strategy. This is achieved by constructing the shortest patrol schedule that is equivalent to the compact strategy, and comparing the patrol travel time against τ.

Let $S(n)$ represent all the compact strategies that contain n patrol areas. If $S(\check{n})$ contains all the compact strategies that are covered with the highest quality defensive activity at each patrol area, the process of generating compact strategies will terminate and $\check{n}$ is called the stopping point of enumeration. Any compact strategy that contains fewer than $\check{n}$ patrol areas will be dominated by a compact strategy in $S(\check{n})$.

Figure 2 shows a high-level view of the steps of the algorithm using the compact representation. The compact strategies are used instead of full patrol schedules to generate the game matrix. Once the optimal probability distribution is calculated (as explained in Section 4.3) for the compact strategies, the strategies with nonzero probability are expanded to a complete set of patrol schedules.

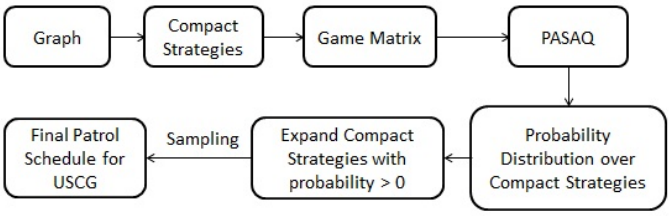


Fig. 2 Flow chart of the PROTECT system.

In this expansion from a compact strategy to a full set of patrol schedules, we need to determine the probability of choosing each patrol schedule, since a compact strategy may correspond to multiple patrol schedules. The focus here is on increasing the difficulty for the attacker of conducting surveillance by increasing unpredictability,[2] which we achieve by randomizing uniformly over all expansions of the compact defender strategies. The uniform distribution provides the maximum entropy (greatest unpredictability). Thus, all the patrol schedules generated from a single compact strategy are assigned a probability of v_i/w_i where v_i is the probability of choosing a compact strategy Γ_i and w_i is the total number of expanded patrol schedules for Γ_i. The complete set of patrol schedules and the associated probabilities are then sampled and provided to the USCG, along with the start time of the patrol generated via uniform random sampling.

4.3 Human Adversary Modeling

While previous game-theoretic security applications have assumed a perfectly rational attacker, PROTECT takes a step forward by addressing this limitation of classical game theory. Instead, PROTECT permits a boundedly rational adversary, using a quantal response (QR) adversary reasoning model, which has shown to be a promising model of human decision-making [14, 18, 25]. A recent study demonstrated the use of QR as an effective prediction model of humans [24]. An even more relevant study of the QR model was conducted by Yang et al. [25] in the context of security games where this model was shown to outperform competitors in modeling human subjects. Based on this evidence, PROTECT uses a QR model of a human adversary. (Aided by a software assistant, the defender still computes the optimal mixed strategy.)

The QR model adapts ideas from the literature which presumes that humans will choose better actions at a higher frequency, but with noise added to the decision-making process following a logit distribution:

[2] Creating optimal Stackelberg defender strategies that increase the attacker's difficulty of surveillance is an open research issue in the literature; here we choose to maximize unpredictability as the first step.

$$q_i = \frac{e^{\lambda G_i^a(x_i)}}{\sum_{j=1}^{T} e^{\lambda G_j^a(x_i)}} \tag{2}$$

The parameter $\lambda \in [0, \infty]$ represents the amount of noise in the attacker's strategy, with a value of 0 indicating a uniform random probability over attacker strategies and a value of ∞ indicating a perfectly rational attacker. q_i is the probability that the attacker chooses target i; $G_i^a(x_i)$ is the attacker's expected utility of attacking target i given x_i, the probability that the defender covers target i; and T is the total number of targets.

To apply the QR model within a Stackelberg framework, PROTECT employs an algorithm known as PASAQ [26]. PASAQ computes the optimal defender strategy (within a guaranteed error bound) given a QR model of the adversary by solving the following nonlinear, non-convex optimization problem P (see Table 4 for notation):

$$\mathtt{P}: \begin{cases} \max\limits_{x,a} \dfrac{\sum_{i=1}^{T} e^{\lambda(R_i^a - (R_i^a - P_i^a)x_i)}((R_i^d - P_i^d)x_i + P_i^d)}{\sum_{i=1}^{T} e^{\lambda(R_i^a - (R_i^a - P_i^a)x_i)}} \\ x_i = \sum\limits_{j=1}^{J} a_j A_{ij}, \quad \forall i \\ \sum\limits_{j=1}^{J} a_j = 1 \\ 0 \le a_j \le 1, \quad \forall j \end{cases}$$

t_i	target i
R_i^d	defender reward for covering t_i if attacked
P_i^d	defender penalty for not covering t_i if attacked
R_i^a	attacker reward for attacking t_i if not covered
P_i^a	attacker penalty for attacking t_i if covered
A_{ij}	effectiveness probability of compact strategy Γ_j on t_i
a_j	probability of choosing compact strategy Γ_j
J	total number of compact strategies
x_i	marginal coverage on t_i

Table 4 PASAQ notation as applied to PROTECT.

The first line of the problem corresponds to the computation of the defender's expected utility resulting from a combination of Equations 1 and 2. Unlike previous applications [11, 21], x_i in this case not just summarizes presence or absence on a target, but also the effectiveness probability A_{ij} on the target as well.

The key idea in PASAQ is to use a piecewise linear function to approximate the nonlinear objective function appearing in formulation P, and thus convert it into a Mixed-Integer Linear Programming (MILP) problem, which can then be solved in a reasonable amount of time. Such a problem can easily include assignment constraints giving an approximate solution for a Stackelberg game against a QR-

adversary with assignment constraints, as is the case in this setting. See [26] for details.

As with all QR models, a value for λ is needed to represent the noise in the attacker's strategy. Based on discussions with USCG experts about the attacker's behavior, a λ value of 0 (uniform random) and ∞ (fully rational) were ruled out. Under the payoff data for Boston, an attacker's strategy with $\lambda = 4$ starts approaching a fully rational attacker—the probability of attack focuses on a single target. It was determined from the information gathered from the USCG that the attacker's strategy is best modeled with a value $\lambda \in [0.5, 4]$. A discrete sampling approach was used to determine a λ value giving the highest average expected utility across attacker strategies within this range, yielding $\lambda = 1.5$. Selecting an appropriate value for λ remains a complex issue, however, and it is a key agenda item for future work.

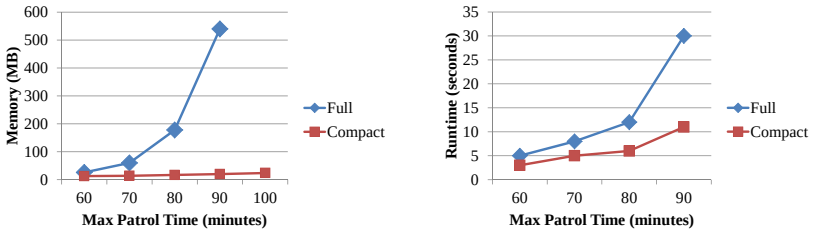
5 Evaluation

This section presents evaluations based on (i) experiments completed via simulations and (ii) real-world patrol data along with USCG analysis. All scenarios and experiments, including the payoff values and graph (composed of 9 patrol areas), were based on the port of Boston. The defender's payoff values lie in the range $[-10, 5]$ while the attacker's payoff values lie in the range of $[-5, 10]$. The game was modeled as a zero-sum game[3] in which the attacker's loss or gain is balanced precisely by the defender's gain or loss. For PASAQ, the defender's strategy uses $\lambda = 1.5$ as mentioned in Section 4.3. All experiments are run on a machine with an Intel Dual Core 1.4 GHz processor and 2 GB of RAM.

5.1 Memory and Run-time Analysis

This section presents the results based on simulation to show the efficiency in memory and run-time of the compact representation versus the full representation (Section 4.2). In Figure 3(a), the x and y axes indicate the maximum patrol time allowed and the memory needed to run PROTECT, respectively. In Figure 3(b), the x and y axes indicate the maximum patrol time allowed and the run-time of PROTECT, respectively. The maximum patrol time allowed determines the number of combinations of patrol areas that can be visited; that is, the x-axis indicates a scale-up in the number of defender strategies. When the maximum patrol time is set to 90 minutes, the full representation takes 30 seconds and uses 540 MB of memory while the compact representation takes 11 seconds to run and requires 20 MB of memory. Due to the exponential increase in the memory and run-time for the full representation, it cannot feasibly be scaled up beyond 90 minutes in our simulation setting.

[3] In general these types of security games are non-zero-sum [21], though for Boston as a first step it was decided to cast the game as zero-sum.



(a) Memory comparison (b) Runtime comparison

Fig. 3 Comparison of full vs. compact representation.

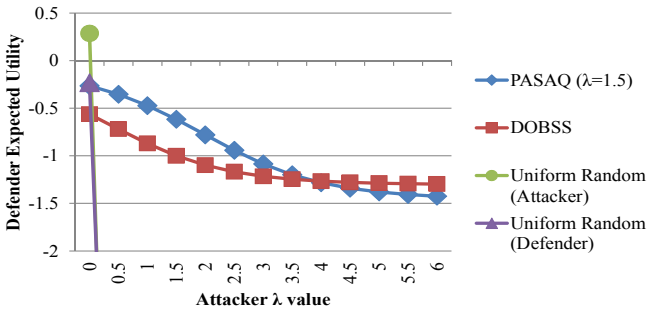


Fig. 4 Defender's expected utility when varying λ for attacker's strategy.

5.2 Utility Analysis

Since we are working with real data, it is interesting to understand whether PROTECT using PASAQ with $\lambda = 1.5$ provides an advantage over (i) a uniform random defender's strategy; (ii) a mixed strategy assuming the attacker chooses targets uniformly at random ($\lambda = 0$), and (iii) a mixed strategy assuming a fully rational attacker ($\lambda = \infty$). The existing DOBSS algorithm was used for $\lambda = \infty$ [21]. Additionally, the $\lambda = \infty$ setting provides an interesting comparison because of its extensive use in previous applications. (For our zero-sum case, DOBSS is equivalent to minimax but the utility does not change.) In typical settings we might not have a reliable estimate of the exact value of λ, but only an estimated range. Therefore, ideally we wish to show that PROTECT (using PASAQ with $\lambda = 1.5$) provides an advantage over a range of λ values, not just over a point estimate.

To achieve this, we compute the average defender utility of the four approaches above as λ varies over the range $[0,6]$, which is a more conservative range than $[0.5,4]$. In Figure 4, the y-axis indicates the defender's expected utility; the x-axis indicates the λ value used for the attacker's strategy. Both uniform random strategies perform well when the attacker's strategy is based on $\lambda = 0$. However, as λ increases, both strategies quickly drop to a very low defender expected utility. In contrast, PASAQ with $\lambda = 1.5$ provides a higher expected utility than that it does

assuming a fully rational attacker over a range of attacker λ values (and indeed over the range of interest), not just at $\lambda = 1.5$.

5.3 Robustness Analysis

In the real world, observation, execution, and payoffs are not always perfect due to various causes: noise in the attacker's surveillance of the defender's patrols, the many tasks and responsibilities of the USCG, whose crews may be pulled off a patrol, and limited knowledge of the attacker's payoff values. Our hypothesis is that PASAQ with $\lambda = 1.5$ is more robust to such noise than a defender strategy (such as DOBSS [21]) that assumes full attacker rationality; that is, we believe PASAQ's expected defender utility will be more robust than DOBSS' over the range of attacker λ of interest. This is illustrated by comparing both PASAQ and DOBSS against observation, execution, and payoff noise [11, 13, 27]. (A comparison of the uniform random strategies was omitted due to its poor performance shown in Figure 4.) All experiments were run generating 200 samples with added noise, averaging over all the samples. In Figures 5, 6, and 7, the y-axis indicates the defender's expected utility, and the x-axis indicates the attacker's λ value, with error bars depicting the standard error.

The first experiment considers observational noise, which means the attacker has noise associated with observing the defender's patrol strategy, as shown in Figure 5. In this scenario, if the defender covers a target with probability p, the attacker may perceive the probability to be uniformly distributed in $[p-x, p+x]$ where x is the noise value. The low observation error corresponds to $x = 0.1$, and the high error to $x = 0.2$. Contrary to expectation, we find that observation error leads to an increase in defender expected utility in PASAQ, but a potential decrease (or no change) in DOBSS. Thus PASAQ ends up dominating DOBSS by a larger margin over bigger ranges of λ values, further strengthening the case for using PASAQ rather than a full-rationality model.

An example illustrates PASAQ's unexpected behavior. Suppose the defender's strategy is $\mathbf{c}$ and there are two targets t_1 and t_2 with defender expected utilities of $U_1^d(\mathbf{c}) = -2$ and $U_2^d(\mathbf{c}) = -1$ (with the attacker's expected utility $U^a(\mathbf{c})$ being the opposite, since the game is zero-sum). For an attacker strategy with a larger λ, the adversary will choose to attack t_1 and the defender would receive a utility of -2. When observation noise is added, increases in the coverage of t_1 results in decreases in $U_1^a(\mathbf{c}')$, so the attacker might choose to attack t_2 instead, giving the defender a greater utility. If the coverage of t_1 decreases, $U_1^a(\mathbf{c}')$ will increase and the attacker will still choose to attack t_1, but $U_1^d(\mathbf{c})$ will remain the same as when there was no noise.

DOBSS exhibits a different trend because DOBSS minimizes the maximum attacker's expected utility or, in our zero-sum setting, maximizes the minimum defender's expected utility. This results in multiple targets with the same minimum defender's utility, which are referred to as an *attack set* [21]. Typically, when the

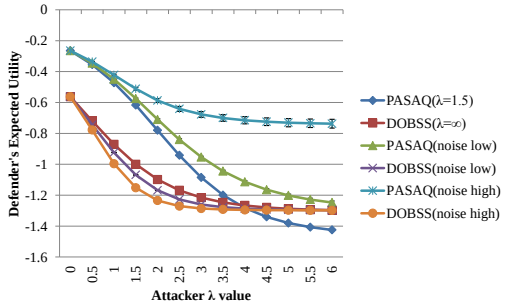


Fig. 5 Defender's expected utility: observation noise.

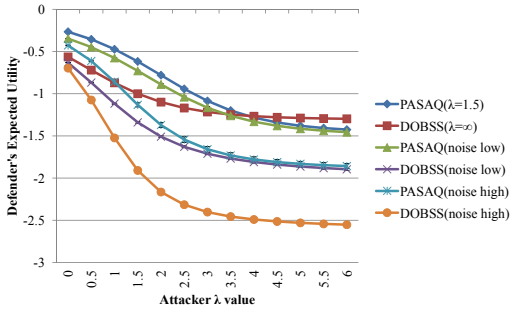


Fig. 6 Defender's expected utility: execution noise.

coverage over the attack set varies due to observation error, some of the targets have less coverage and some have more, but the attacker ends up attacking the targets in the attack set regardless, giving the defender almost no change in its expected utility.

For the second experiment, noise is added to the execution phase of the defender, as shown in Figure 6. If the defender covered a target with probability p, this probability now changes to be uniformly distributed within $[p-x, p+x]$, where x is the noise value. The low execution error considered is $x = 0.1$, and the high error is $x = 0.2$. The key takeaway here is that execution error leads to PASAQ dominating DOBSS over all tested values of λ, further strengthening the reason to use PASAQ rather than a full-rationality model. When execution error is added, PASAQ dominates DOBSS because the latter seeks to maximize the minimum defender's expected utility, and so multiple targets will have the same minimum defender utility. For DOBSS, when execution error is added there is a greater probability that one of these targets will have less coverage, resulting in a lower defender's expected utility. For PASAQ, typically only one target has the minimum defender expected utility. As a result, changes in coverage do not impact it as much as DOBSS. As with observation error, as execution error increases, the advantage of PASAQ over DOBSS for the defender's expected utility grows even greater.

In the third experiment, shown in Figure 7, payoff noise is added by aggregating mean-0 Gaussian noise to the attacker's original payoff values (similar to [11]). As more noise is added to the payoffs, both defenders' strategies result in an increase

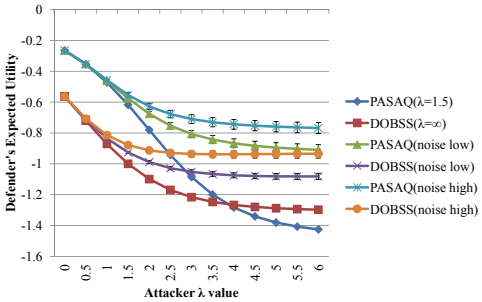


Fig. 7 Defender's expected utility: payoff noise.

in the defender's expected utility because the game is no longer zero-sum. The low payoff noise corresponds to a standard deviation of 1 while a high payoff noise corresponds to a standard deviation of 1.5. Similar to the previous experiments, when payoff noise is added, PASAQ again dominates DOBSS. As noise is added to the attacker's payoff but *not* the defender's payoff, the attacker's strategy may no longer yield the lowest possible defender expected utility. For example, with no payoff noise, target t_1 gives the attacker the highest utility and the defender the lowest utility. When noise is added to the attacker's payoffs, t_1 may no longer give the attacker the highest utility; instead, he/she will choose to attack target t_2, and the defender receives a higher utility than t_1. In essence, with a zero-sum game the defender plans a conservative strategy, based on maximin, and as such any change in the attacker is to the defender's benefit.

5.4 USCG Real-World Evaluation

In addition to the data made available from simulations, the USCG conducted its own real-world evaluation of PROTECT. With permission, some aspects of the evaluation are presented in this chapter.

Real-world scheduling data: A key novelty of this chapter is the inclusion of actual data from USCG patrols before and after the deployment of PROTECT at the port of Boston. Only recently has there been real-world data used in game theoretical security applications. A previous application by Shakarian et al. [19] used real data about Improvised Explosive Device (IED) attacks, to predict locations that the adversary may try to hide. Our application and use of real world data is different in that it shows and compares actual patrols conducted by USCG personnel before and after PROTECT.

Figure 8 and Figure 9 present the frequency of visits by USCG to different patrol areas over a number of weeks. The x-axis indicates day of the week, and the y-axis indicates the number of times a patrol area is visited on that day. The y-axis is intentionally blurred for security reasons, since this is real data from Boston. There are more lines in Figure 8 than in Figure 9 because during the implementation of

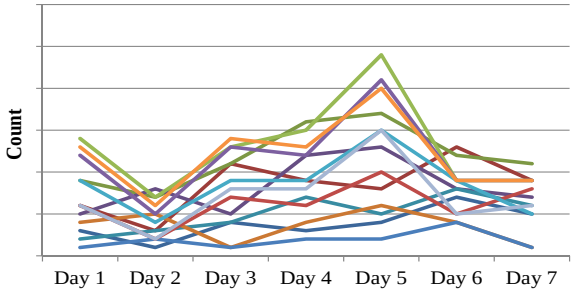


Fig. 8 Patrol visits per day by area – pre-PROTECT.

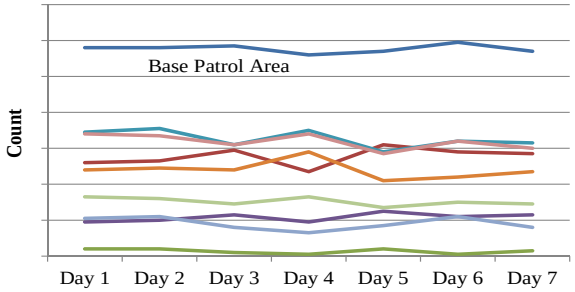


Fig. 9 Patrol visits per day by area – post-PROTECT.

PROTECT, new patrol areas were formed, containing more targets and thus fewer patrol areas. Figure 8 depicts a definite pattern in the patrols. While there is a spike in patrols executed on Day 5, there is a dearth of patrols on Day 2. Besides this pattern, the lines in Figure 8 intersect, indicating that on some days a higher value target was visited more often while on other days it was visited less often, even though the value of a target does not change day-to-day. This means that there was not a consistently high frequency of coverage of higher value targets before PROTECT.

In Figure 9, we notice that the pattern of low patrols on Day 2 (from Figure 8) disappears. Furthermore, lines do not frequently intersect, that is, higher-value targets are visited consistently across the week. The top line in Figure 9 is the base patrol area and is necessarily visited more often than the other patrol areas.

Adversary Perspective Teams (APT): To obtain a better understanding of how the adversary views the potential targets in the port, the USCG created an Adversarial Perspective Team (APT), a mock attacker team. The APT provides assessments from the terrorist perspective and as a secondary function, assesses the effectiveness of the patrol activities before and after deployment of PROTECT. In their evaluation, the APT incorporates the adversary's known intent, capabilities, skills, commitment, resources, and cultural influences. In addition, it screens attack possibilities and assists in identifying the level of deterrence projected at and perceived by the adversary. For the purposes of this research, the adversary is defined as individuals with ties to al-Qa'ida or its affiliates.

The APT conducted a pre- and post-PROTECT assessment of the system's impact on an adversary's deterrence at the port of Boston. This analysis uncovered a positive trend in which the deterrence effectiveness increased between the pre- to post- PROTECT observations.

Additional Real-world Indicators: The use of PROTECT and the APT's improved guidance given to boat crews on how to conduct patrols jointly provided a noticeable increase in the quality and effectiveness of the patrols. Prior to implementing PROTECT, there were no documented reports of illicit activity. After implementation, USCG crews, reported more illicit activities within the port and provided a noticeable "on the water" presence with industry port partners commenting, "the Coast Guard seems to be everywhere, all the time." With no actual increase in the number of resources applied, and therefore no increase in capital or operating costs, these outcomes support the practical application of game theory in the maritime security environment.

5.5 Outcomes following the Boston Implementation

After evaluating the performance and impact of PROTECT at Boston, the USCG viewed this system as a success. As a result, PROTECT is now being deployed in the port of New York, a much larger and more complicated security environment. We were presented with an award for the work on the PROTECT system for Boston Harbor, which reflects the USCG's recognition of PROTECT's impact and value.

6 Lessons Learned: Putting Theory into Practice

The development of the PROTECT model was a collaborative effort involving university researchers and USCG personnel, including decision-makers, planners and operators. Building on the lessons reported in [21] for working with security organizations, we informed the USCG of (i) the assumptions underlying the game-theoretic approaches, e.g. full adversary rationality, and strengths and limitations of different algorithms, rather than pre-selecting a simple heuristic approach; (ii) the need to define and collect correct inputs for model development and; (iii) a fundamental understanding of how the inputs affect the results. We gained new insights on real-world applied research, in particular involving the following issues: (i) unforeseen positive benefits that can occur when security agencies are compelled to reexamine their assumptions; (ii) the requirement to work with multiple teams in a security organization at multiple levels of their hierarchy and; (iii) the necessity of preparing answers to practical end-user questions not always directly related to "meaty" research problems.

The first insight came about when the USCG had to reassess their operational assumptions as a result of working through this research problem. A positive result

of this reexamination prompted the USCG to develop new PWCS mission tactics, techniques, and procedures. Through an iterative development process, the USCG reassessed the reasons why boat crews performed certain activities and whether they were sufficient. For example, instead of "covered" vs. "not covered" as the only two possibilities at a patrol point, there are now multiple sets of possible activities for each patrol point.

The second insight is that applied research requires the research team to collaborate with planners and operators on the multiple levels of a security organization to ensure that the model accounts for all aspects of a complex real-world environment. Initially when we started working on PROTECT, the focus was on patrolling each individual target. This appeared to micromanage the activities of boat crews, and it was through their input that individual targets were grouped into patrol areas associated with a PWCS patrol. On the other hand, input from USCG headquarters and the APT mentioned earlier led to other changes in PROTECT, e.g. departing from a fully rational adversary model to a QR model.

The third insight involves the need to develop answers to end-user questions that are not always related to "meaty" research questions but are related to the larger knowledge domain on which the research depends. One example of this involved the user citing that one patrol area was being repeatedly visited, which seemed to suggest the schedules were not genuinely random. After assessing this concern, we determined that the cause of the repeated visits to the patrol area was its high reward, an order of magnitude greater than those of the rarely visited patrol areas. PROTECT correctly assigned patrol schedules that covered the more "important" patrol areas more frequently. In another example, the user noted that PROTECT did not assign any patrols to start at 4:00 AM or 4:00 PM over a 60 day test period. They expected patrols would be scheduled to start at any hour of the day, prompting them to ask if there was a problem with the program. This required us to develop a layman's briefing on probabilities, randomness, and sampling. With 60 patrol schedules, a few start hours may not be chosen given our uniform random sampling of the start time. These practitioner-based issues demonstrate the need for researchers to not only be conversant in the algorithms and mathematics underlying the research, but also be able to explain from a user's perspective why solutions make sense. An inability to address these issues would result in a lack of real-world user confidence in the model.

7 Summary and Related Work

This chapter reports on PROTECT, a game-theoretic system deployed by the USCG in the port of Boston since April 2011 for scheduling their patrols. USCG has deemed the deployment of PROTECT in Boston a success and efforts are underway to deploy PROTECT in the port of New York, and to other ports in the United States. PROTECT uses an attacker-defender Stackelberg game model, and includes five key innovations.

First, PROTECT moves away from the assumption of perfect adversary rationality used in previous work, relying instead on a quantal response (QR) model of the adversary's behavior. While the QR model has been studied extensively in behavioral game theory, to the best of our knowledge this is its first real-world deployment. Second, to improve PROTECT's efficiency, we generate a novel compact representation of the defender's strategy space, exploiting equivalence and dominance. Third, the paper shows how to practically model a real-world (maritime) patrolling problem as a Stackelberg game. Fourth, we provide experimental results illustrating that PROTECT's QR model of the adversary is better able to handle real-world uncertainties than a perfect rationality model can. Finally, for the first time in a security application evaluation, we use real-world data, providing (i) a comparison of human-generated security schedules and those generated via a game-theoretic algorithm, and (ii) results from an APT's analysis of the impact of the PROTECT system. We also outlined insights gained from the project, which include the ancillary benefits due to a review of assumptions made by security agencies and the importance of answering questions not directly related to the research problem.

As a result, PROTECT has advanced the state of the art beyond previous applications of game theory for security. Prior applications mentioned earlier, including ARMOR, IRIS or GUARDS [21], have each provided unique contributions in applying novel game-theoretic algorithms and techniques. Interestingly, these applications have revolved around airport and air-transportation security. PROTECT's novelty is not only its application domain in maritime patrolling, but also in the five key innovations mentioned above, particularly the moving away from the assumption of perfect rationality by using the QR model.

In addition to game-theoretic applications, the issue of patrolling has received significant attention in the multi-agent literature. These include patrol work done by robots primarily for perimeter patrols that have been addressed in arbitrary topologies [2], maritime patrols in simulations for deterring pirate attacks [23], and in research on the impact of uncertainty in adversarial behavior [1]. PROTECT differs from these approaches in its use of a QR model of a human adversary in a game-theoretic setting, and in being a deployed application. Building on this initial success of PROTECT, we hope to deploy it at more and much larger-sized ports. In so doing, in the future, we will consider significantly more complex attacker strategies, including potential real-time surveillance and coordinated attacks.

8 Acknowledgments

We thank the USCG offices, and particularly Sector Boston, for their exceptional collaboration. Thanks to Matt Johnson for technical assistance in the preparation of this chapter. The views expressed herein are those of the author(s) and are not to be construed as official or reflecting the views of the Commandant or of the U.S. Coast Guard. This research was supported by the United States Department of Homeland

Security through the Center for Risk and Economic Analysis of Terrorism Events (CREATE) under award number 2010-ST-061-RE0001.

References

1. N. Agmon, S. Kraus, G. A. Kaminka, and V. Sadov. Adversarial uncertainty in multi-robot patrol. In *IJCAI*, 2009.
2. N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *AAMAS*, 2009.
3. D. Blair. Annual threat assessment of the US intelligence community for the senate select committee on intelligence. *http://www.dni.gov/testimonies/20100202_testimony.pdf*, 2010.
4. C. F. Camerer. *Behavioral Game Theory: Experiments in Strategic Interaction*. Princeton University Press, 2003.
5. V. Conitzer and T. Sandholm. Computing the optimal strategy to commit to. In *ACM EC*, 2006.
6. K. Dozier. Bin laden trove of documents sharpen US aim. *http://www.msnbc.msn.com/id/43331634/ns/us_news-security/t/bin-laden-trove-documents-sharpen-us-aim/*, 2011.
7. D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 1991.
8. M. Jain, E. Kardes, C. Kiekintveld, F. Ordonez, and M. Tambe. Security games with arbitrary schedules: A branch and price approach. In *AAAI*, 2010.
9. M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, and F. Ordonez. Software assistants for randomized patrol planning for the LAX airport police and the federal air marshal service. *Interfaces*, 40:267–290, 2010.
10. C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordóñez. Computing optimal randomized resource allocations for massive security games. In *AAMAS*, 2009.
11. C. Kiekintveld, J. Marecki, and M. Tambe. Approximation methods for infinite bayesian Stackelberg games: modeling distributional uncertainty. In *AAMAS*, 2011.
12. D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Proc. of The 24th AAAI Conference on Artificial Intelligence*, pages 805–810, 2010.
13. D. Korzhyk, V. Conitzer, and R. Parr. Solving Stackelberg games with uncertain observability. In *AAMAS*, 2011.
14. R. D. McKelvey and T. R. Palfrey. Quantal response equilibria for normal form games. *Games and Economic Behavior*, 10(1):6–38, 1995.
15. P. Paruchuri, J. P. Pearce, J. Marecki, M. Tambe, F. Ordonez, and S. Kraus. Playing games with security: An efficient exact algorithm for Bayesian Stackelberg games. In *AAMAS*, 2008.
16. J. Pita, M. Jain, C. Western, C. Portway, M. Tambe, F. Ordonez, S. Kraus, and P. Parachuri. Deployed ARMOR protection: The application of a game-theoretic model for security at the Los Angeles International Airport. In *AAMAS*, 2008.
17. J. Pita, M. Tambe, C. Kiekintveld, S. Cullen, and E. Steigerwald. GUARDS - game theoretic security allocation on a national scale. In *AAMAS*, 2011.
18. B. W. Rogers, T. R. Palfrey, and C. F. Camerer. Heterogeneous quantal response equilibrium and cognitive hierarchies. *Journal of Economic Theory*, 2009.
19. P. Shakarian, J. P. Dickerson, and V. S. Subrahmanian. Adversarial geospatial abduction problems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 3(2), 2012.
20. E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, and G. Meyer. Protect: A deployed game theoretic system to protect the ports of the united states. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*, 2012.
21. M. Tambe. *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned*. Cambridge University Press, 2011.

22. J. Tsai, S. Rathi, C. Kiekintveld, F. Ordonez, and M. Tambe. IRIS: a tool for strategic security allocation in transportation networks. In *AAMAS*, pages 37–44, 2009.
23. O. Vanek, M. Jakob, O. Hrstka, and M. Pechoucek. Using multi-agent simulation to improve the security of maritime transit. In *MABS*, 2011.
24. J. Wright and K. Leyton-Brown. Beyond equilibrium: Predicting human behavior in normal form games. In *AAAI*, 2010.
25. R. Yang, C. Kiekintveld, F. Ordonez, M. Tambe, and R. John. Improving resource allocation strategy against human adversaries in security games. In *IJCAI*, 2011.
26. R. Yang, M. Tambe, and F. Ordonez. Computing optimal strategy against quantal response in security games. In *AAMAS*, 2012.
27. Z. Yin, M. Jain, M. Tambe, and F. Ordóñez. Risk-averse strategies for security games with execution and observational uncertainty. In *AAAI*, 2011.