Chapter 16
Towards a Science of Security Games

Thanh Hong Nguyen, Debarun Kar, Matthew Brown, Arunesh Sinha, Albert Xin Jiang, and Milind Tambe

Abstract Security is a critical concern around the world. In many domains from counter-terrorism to sustainability, limited security resources prevent full security coverage at all times; instead, these limited resources must be scheduled, while simultaneously taking into account different target priorities, the responses of the adversaries to the security posture and potential uncertainty over adversary types.

Computational game theory can help design such security schedules. Indeed, casting the problem as a Bayesian Stackelberg game, we have developed new algorithms that are now deployed over multiple years in multiple applications for security scheduling. These applications are leading to real-world use-inspired research in the emerging research area of “security games”; specifically, the research challenges posed by these applications include scaling up security games to large-scale problems, handling significant adversarial uncertainty, dealing with bounded rationality of human adversaries, and other interdisciplinary challenges.

Keywords Security games • Bayesian Stackelberg games • Game theory • Scalability • Uncertainty • Bounded rationality

16.1 Introduction

Security is a critical concern around the world that arises in protecting our ports, airports, transportation and other critical national infrastructure from adversaries, in protecting our wildlife and forests from poachers and smugglers, and in curtailing the illegal flow of weapons, drugs, and money; and it arises in problems ranging from physical to cyber-physical systems. In all of these problems, we have limited security resources which prevent full security coverage at all times; instead, security

T.H. Nguyen (✉) • D. Kar • M. Brown • A. Sinha • M. Tambe
University of Southern California, Los Angeles, CA, USA
e-mail: thanhhng@usc.edu; dkar@usc.edu; mattheab@usc.edu; aruneshs@usc.edu; tambe@usc.edu

A.X. Jiang
Trinity University, San Antonio, TX, USA
e-mail: xjiang@trinity.edu

B. Toni (ed.), *Mathematical Sciences with Multidisciplinary Applications*, Springer Proceedings in Mathematics & Statistics 157, DOI 10.1007/978-3-319-31323-8_16

resources must be deployed intelligently taking into account differences in priorities of targets requiring security coverage, the responses of the attackers to the security posture, and potential uncertainty over the types, capabilities, knowledge, and priorities of attackers faced.

Game theory, which studies interactions among multiple self-interested agents, is well-suited to the adversarial reasoning required for security resource allocation and scheduling problems. Casting the problem as a Bayesian Stackelberg game, we have developed new algorithms for efficiently solving such games that provide randomized patrolling or inspection strategies. These algorithms have led to some initial successes in this challenging problem arena, leading to advances over previous approaches in security scheduling and allocation, e.g., by addressing key weaknesses of predictability of human schedulers. These algorithms are now deployed in multiple applications: ARMOR has been deployed at the Los Angeles International Airport (LAX) since 2007 to randomize checkpoints on the roadways entering the airport and canine patrol routes within the airport terminals [17]; IRIS, a game-theoretic scheduler for randomized deployment of the US Federal Air Marshals Service (FAMS) requiring significant scale-up in underlying algorithms, has been in use since 2009 [17]; PROTECT, which schedules the US Coast Guard's (USCG) randomized patrolling of ports using a new set of algorithms based on modeling bounded-rational human attackers, has been deployed in the port of Boston since April 2011 and is in use at the port of New York since February 2012 [39], and is headed for nationwide deployment; another application for deploying escort boats to protect ferries has been deployed by the USCG since April 2013 [10]; and TRUSTS [51] has been evaluated in field trials by the Los Angeles Sheriffs Department (LASD) in the LA Metro system and a nationwide deployment is now being evaluated at TSA. Most recently, PAWS—another game-theoretic application using a Bayesian distribution of boundedly rational attackers was tested by rangers in Uganda for protecting wildlife in Queen Elizabeth National Park (QENP) in April 2014 [49]; MIDAS which is based on modeling behaviors of attackers combined with the robust approach is in use by USCG for protecting fisheries [14]. These initial successes point the way to major future applications in a wide range of security domains; with major research challenges in scaling up our game-theoretic algorithms, in addressing human adversaries' bounded rationality and uncertainties in action execution and observation, as well as in multiagent learning.

Given many game-theoretic applications for solving real-world security problems, this book chapter will provide an overview of the models and algorithms, key research challenges and a brief description of our successful deployments with emphasis on *three key lessons*: (1) computational game theory-based decision aids are in daily use by security agencies due to their capability for optimizing limited security resources against strategic adversaries; (2) these applications provide fundamental research challenges, leading to an (emerging) science of security games, including the challenge of massive scale games which cannot fit into memory and the challenge of modeling many different forms of uncertainty in outcomes and preferences, action execution, and human decision-making; and (3) current security game applications for solving *green security games* such as protecting wildlife and the environment are challenging for AI; these are important global problems that provide open research problems to integrate AI research (including planning and learning) in security games.

16.2 Stackelberg Security Games

Stackelberg security games (SSGs) were first introduced to model leadership and commitment [44], and are now used to study security problems ranging from "police and robbers" scenario [12], computer network security [29], missile defense systems [5], and terrorism [38]. Models for arms inspections and border patrolling have also been modeled using inspection games [3], a related family of Stackelberg games.

This section provides details on this use of Stackelberg games for modeling security domains. We first give a generic description of security domains followed by *security games*, the model by which security domains are formulated in the Stackelberg game framework.

16.2.1 Security Domain Description

In a security domain, a defender must perpetually defend a set of targets using a limited number of resources, whereas the attacker is able to surveil and learn the defender's strategy and attack after careful planning. This fits precisely into the description of a Stackelberg game if we map the defender to the leader's role and the attacker to the follower's role [3, 6]. An action, or *pure strategy*, for the defender represents deploying a set of resources on patrols or checkpoints, e.g., scheduling checkpoints at the LAX airport or assigning federal air marshals to protect flight tours. The pure strategy for an attacker represents an attack at a target, e.g., a flight. The strategy for the leader is a *mixed strategy*, a probability distribution over the pure strategies of the defender. Additionally, with each target are also associated a set of payoff values that define the utilities for both the defender and the attacker in case of a successful or a failed attack. These payoffs are represented using the *security game* model, described next.

16.2.2 Definition of SSGs

A key assumption of security games is that the payoff of an outcome depends only on the target attacked, and whether or not it is covered by the defender [25]. The payoffs do *not* depend on the remaining aspects of the defender allocation.

Table 16.1 Example of a security game with two targets

	Defender		Attacker	
Target	Covered	Uncovered	Covered	Uncovered
t_1	10	0	−1	1
t_2	0	−10	−1	1

For example, if an adversary succeeds in attacking target t_1, the penalty for the defender is the same whether the defender was guarding target t_2 or not.

This allows us to compactly represent the payoffs of a security game. Specifically, a set of four payoffs is associated with each target. These four payoffs are the rewards and penalties to both the defender and the attacker in case of a successful or an unsuccessful attack, and are sufficient to define the utilities for both players for all possible outcomes in the security domain. Table 16.1 shows an example security game with two targets: t_1 and t_2. In this example game, if the defender was *covering* (protecting) target t_1 and the attacker attacked t_1, the defender would get 10 units of reward whereas the attacker would receive −1 units. We make the assumption that in a security game it is always better for the defender to cover a target as compared to leaving it uncovered, whereas it is always better for the attacker to attack an uncovered target. This assumption is consistent with the payoff trends in the real-world. A special case is *zero-sum games*, in which for each outcome the sum of utilities for the defender and attacker is zero, although in general security games are not necessarily zero-sum.

In the above example, all payoff values are exactly known. In practice, we often have uncertainty over the payoffs and preferences of the players. Bayesian games are a well-known game-theoretic model in which such uncertainty is modeled using multiple types of players, with each associated with its own payoff values. For security games of interest, the main source of payoff uncertainty is regarding the attacker's payoffs. In the resulting *Bayesian Stackelberg game* model, there is only one leader type (e.g., only one police force), although there can be multiple follower types (e.g., multiple attacker types trying to infiltrate security) [35]. Each follower type is represented using a different payoff matrix. The leader does not know the follower's type, but knows the probability distribution over them. The goal is to find the optimal mixed strategy for the leader to commit to, given that the defender could be facing any of the follower types.

16.2.3 Solution Concept: Strong Stackelberg Equilibrium

The solution to a security game is a mixed strategy for the defender that maximizes the expected utility of the defender, given that the attacker learns the mixed strategy of the defender and chooses a best response for himself. This solution concept is known as a Stackelberg equilibrium [27].

The most commonly adopted version of this concept in related literature is called strong Stackelberg equilibrium (SSE) [4, 9, 35, 45]. An SSE for security games is informally defined as follows (the formal definition of SSE is not introduced for brevity, and can instead be found in [25]):

Definition 1. A pair of strategies form a *SSE* if they satisfy

1. The defender plays a best response, that is, the defender cannot get a higher payoff by choosing any other strategy.
2. The attacker plays a best response, that is, given a defender strategy, the attacker cannot get a higher payoff by attacking any other target.
3. The attacker breaks ties in favor of the leader.

The assumption that the follower will always break ties in favor of the leader in cases of indifference is reasonable because in most cases the leader can induce the favorable strong equilibrium by selecting a strategy arbitrarily close to the equilibrium that causes the follower to strictly prefer the desired strategy [45]. Furthermore an SSE exists in all Stackelberg games, which makes it an attractive solution concept compared to versions of Stackelberg equilibrium with other tie-breaking rules. Finally, although initial applications relied on the SSE solution concept, we have since proposed new solution concepts that are more robust against various uncertainties in the model [1, 37, 50] and have used these robust solution concepts in some of the later applications.

16.3 Deployed Real-World Security Applications

In this section, we describe several deployed and emerging applications of the Stackelberg game framework in different real-world domains. Besides describing successful transitions of research, our aim is to set the stage for later sections in which we discuss the research challenges that arise.

16.3.1 ARMOR for Los Angeles International Airport

Los Angeles International Airport (LAX) is the largest destination airport in the USA and serves 60–70 million passengers per year. The LAX police use diverse measures to protect the airport, which include vehicular checkpoints, police units patrolling the roads to the terminals, patrolling inside the terminals (with canines), and security screening and bag checks for passengers. The application of our game-theoretic approach is focused on two of these measures: (1) placing vehicle checkpoints on inbound roads that service the LAX terminals, including both location and timing, and (2) scheduling patrols for bomb-sniffing canine units at the different LAX terminals. The eight different terminals at LAX have very different

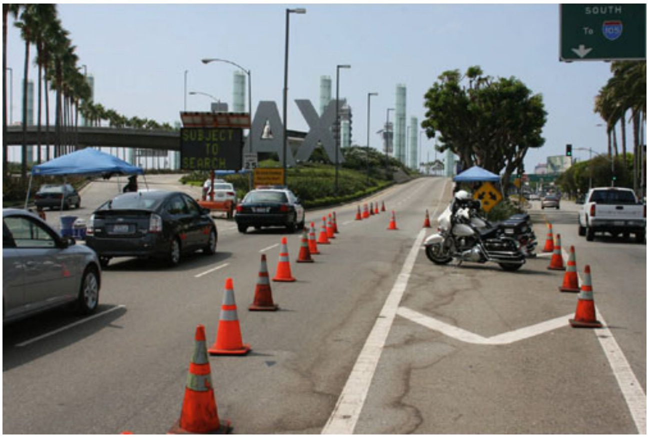


Fig. 16.1 LAX checkpoints are deployed using ARMOR

characteristics, like physical size, passenger loads, international versus domestic flights, etc. These factors contribute to the differing risk assessments of these eight terminals. Furthermore, the numbers of available vehicle checkpoints and canine units are limited by resource constraints. Thus, it is challenging to optimally allocate these resources to improve their effectiveness while avoiding patterns in the scheduled deployments.

The ARMOR system (Assistant for Randomized Monitoring over Routes) focuses on two of the security measures at LAX (checkpoints and canine patrols) and optimizes security resource allocation using Bayesian Stackelberg games. Take the vehicle checkpoints model as an example. Assuming that there are n roads, the police's strategy is placing $m < n$ checkpoints on these roads where m is the maximum number of checkpoints. ARMOR randomizes allocation of checkpoints to roads. The adversary may conduct surveillance of this mixed strategy and may potentially choose to attack through one of these roads. ARMOR models different types of attackers with different payoff functions, representing different capabilities and preferences for the attacker. ARMOR uses DOBSS (Decomposed Optimal Bayesian Stackelberg Solver) [35] to compute the defender's optimal strategy. ARMOR has been successfully deployed since August 2007 at Fig. 16.1.

16.3.2 IRIS for US FAMS

The US FAMS allocates air marshals to flights originating in and departing from the USA to dissuade potential aggressors and prevent an attack should one occur. Flights are of different importance based on a variety of factors such as the numbers of passengers, the population of source and destination, and international flights from different countries. Security resource allocation in this domain is significantly more challenging than for ARMOR: a limited number of air marshals need to be scheduled to cover thousands of commercial flights each day. Furthermore, these air marshals must be scheduled on tours of flights that obey various constraints (e.g., the time required to board, fly, and disembark). Simply finding schedules for the marshals that meet all of these constraints is a computational challenge. Our task is made more difficult by the need to find a randomized policy that meets these scheduling constraints, while also accounting for the different values of each flight.

Against this background, the IRIS system (Intelligent Randomization In Scheduling) has been developed and deployed by FAMS since October 2009 to randomize schedules of air marshals on international flights. In IRIS, the targets are the set of n flights and the attacker could potentially choose to attack one of these flights. The FAMS can assign $m < n$ air marshals that may be assigned to protect these flights. Since the number of possible schedules exponentially increases with the number of flights and resources, DOBSS is no longer applicable to the FAMS domain. Instead, IRIS uses the much faster ASPEN algorithm [16] to generate the schedule for thousands of commercial flights per day.

16.3.3 PROTECT for USCG

The USCG's mission includes maritime security of the US coasts, ports, and inland waterways; a security domain that faces increased risks due to threats such as terrorism and drug trafficking. Given a particular port and the variety of critical infrastructure that an attacker may attack within the port, USCG conducts patrols to protect this infrastructure; however, while the attacker has the opportunity to observe patrol patterns, limited security resources imply that USCG patrols cannot be at every location 24/7. To assist the USCG in allocating its patrolling resources, the PROTECT (Port Resilience Operational/Tactical Enforcement to Combat Terrorism) model has been designed to enhance maritime security. It has been in use at the port of Boston since April 2011, and is also in use at the port of New York since February 2012 (Fig. 16.2). Similar to previous applications ARMOR and IRIS, PROTECT uses an attacker–defender Stackelberg game framework, with USCG as the defender against terrorists that conduct surveillance before potentially launching an attack.

The key idea in PROTECT is also that unpredictability creates situations of uncertainty for an enemy and can be enough to deem a target less appealing. While randomizing patrol patterns is key, PROTECT also addresses the fact that the targets are of unequal value, understanding that the attacker 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 PROTECT builds on previous work, it offers key innovations. First, this system is a departure from the assumption of perfect attacker rationality noted in

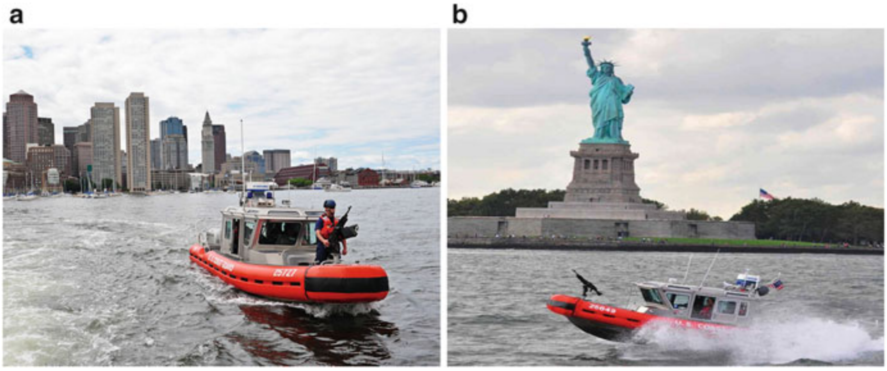


Fig. 16.2 USCG boats patrolling the ports of Boston and NY. (**a**) PROTECT is being used in Boston. (**b**) Extending PROTECT to NY

previous work, relying instead on a quantal response model [31] of the attacker's behavior. Second, to improve PROTECT's efficiency, a compact representation of the defender's strategies is used by exploiting equivalence and dominance. Finally, the evaluation of PROTECT for the first time provides real-world data: (1) comparison of human-generated vs PROTECT schedules, and (2) results from an Adversarial Perspective Team's (APT) (human mock attackers) analysis. The PROTECT model has now been extended to other US ports like Los Angeles/Long Beach and is moving towards nationwide deployment.

16.3.4 Ferry Protection for the USCG

Another problem that USCG faces is the protection of ferries, including the Staten Island Ferry in New York, from potential terrorist attacks from water. We developed a game-theoretic system for scheduling escort boat patrols to protect ferries, and this has been deployed at the Staten Island Ferry since 2013 [10] (Fig. 16.3). The key research challenge is the fact that the ferries are continuous moving in a continuous domain, and the attacker could attack at any moment in time. This type of moving targets domain leads to game-theoretic models with continuous strategy spaces, which presents computational challenges. Our theoretical work showed that while it is safe to discretize the defender's strategy space, discretizing the attacker's strategy space would result in loss of utility. We developed a novel algorithm that uses a compact representation for the defender's mixed-strategy space while being able to exactly model the attacker's continuous strategy space. The implemented algorithm, running on a laptop, is able to generate daily schedules for escort boats with guaranteed expected utility values.

Fig. 16.3 Escort boats protecting the Staten Island Ferry use strategies generated by our system

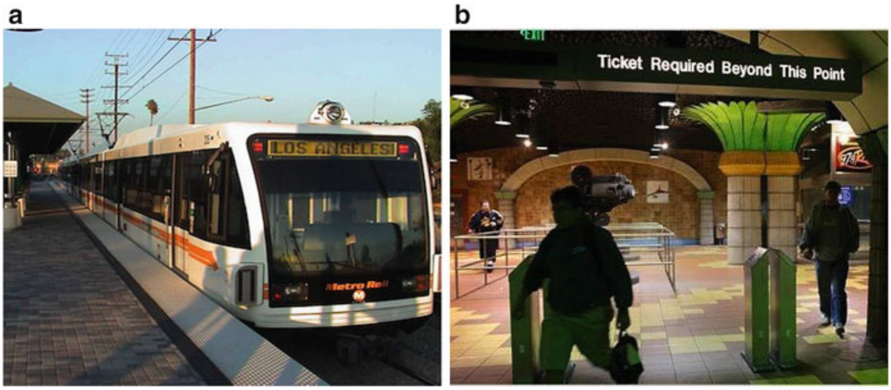


Fig. 16.4 TRUSTS for transit systems. (**a**) Los Angeles Metro. (**b**) Barrier-free entrance to transit system

16.3.5 TRUSTS for Security in Transit Systems

Urban transit systems face multiple security challenges, including deterring fare evasion, suppressing crime and counter-terrorism. In particular, in some urban transit systems, including the Los Angeles Metro Rail system, passengers are legally required to purchase tickets before entering but are not physically forced to do so (Fig. 16.4). Instead, security personnel are dynamically deployed throughout the transit system, randomly inspecting passenger tickets. This proof-of-payment fare collection method is typically chosen as a more cost-effective alternative to direct fare collection, i.e., when the revenue lost to fare evasion is believed to be less than what it would cost to directly preclude it. In the case of Los Angeles Metro, with approximately 300,000 riders daily, this revenue loss can be significant; the annual cost has been estimated at $5.6 million [13]. The Los Angeles Sheriffs Department (LASD) deploys uniformed patrols on board trains and at stations for fare-checking (and for other purposes such as crime prevention). The LASD's current approach relies on humans for scheduling the patrols, which places a tremendous cognitive burden on the human schedulers who must take into account all of the scheduling complexities (e.g., train timings, switching time between trains, and schedule lengths).

The TRUSTS system (Tactical Randomization for Urban Security in Transit Systems) models the patrolling problem as a leader–follower Stackelberg game [51]. The leader (LASD) pre-commits to a mixed-strategy patrol (a probability distribution over all pure strategies), and riders observe this mixed strategy before deciding whether to buy the ticket or not. Both ticket sales and fines issued for fare evasion translate into revenue for the government. Therefore the utility for the leader is the total revenue (total ticket sales plus penalties). The main computational challenge is the exponentially many possible patrol strategies, each subject to both the spatial and temporal constraints of travel within the transit network under consideration. To overcome this challenge, TRUSTS uses a compact representation of the strategy space which captures the spatiotemporal structure of the domain.

The LASD conducted field tests of this TRUSTS system in the LA Metro in 2012, and one of the feedback comments from the officers was that patrols are often interrupted due to execution uncertainty such as emergencies and arrests. Utilizing techniques from planning under uncertainty [in particular Markov Decision Processes (MDPs)], we proposed a general approach to dynamic patrolling games in uncertain environments, which provides patrol strategies with contingency plans [20]. This led to schedules now being loaded onto smartphones and given to officers. If interruptions occur, the schedules are then automatically updated on the smartphone app. The LASD has conducted successful field evaluations using the smartphone app, and the TSA is currently evaluating it towards nationwide deployment.

Crime presents a serious problem in transit systems like LA Metro. Furthermore, unlike terrorists that strategically plans an attack, criminals are often opportunistic, in that their decisions are based on the available opportunities encountered. For the crime problem, we developed a new game-theoretic model that utilizes recent advances in criminology on modeling opportunistic criminals, and novel efficient algorithms that achieve speed-ups by exploiting the spatiotemporal structure of the domain [53].

16.3.6 Fishery Protection for USCG

Fisheries are a vital natural resource from both an ecological and economic standpoint. However, fish stocks around the world are threatened with collapse due to illegal, unreported, and unregulated (IUU) fishing. In the USA, the Coast Guard

(USCG) is tasked with the responsibility of protecting and maintaining the nation's fisheries. To this end, the USCG deploys resources (both air and surface assets) to conduct patrols over fishery areas in order to deter and mitigate IUU fishing. Due to the large size of these patrol areas and the limited patrolling resources available, it is impossible to protect an entire fishery from IUU fishing at all times. Thus, an intelligent allocation of patrolling resources is critical for security agencies like the USCG.

The MIDAS algorithm was developed to address the types challenges faced in natural resource conservation domains such as fishery protection. In stark contrast to counter-terrorism settings, there is frequent interaction between the defender and attacker in these resource conservation domains. This distinction is important for three reasons. First, due to the comparatively low stakes of the interactions, rather than a handful of persons or groups, the defender must protect against numerous adversaries (potentially hundreds or even more), each of which may behave differently. Second, frequent interactions make it possible to collect data on the actions of the adversaries actions over time. Third, the adversaries are less strategic given the short planning windows between actions. Combining these factors, MIDAS models a population of boundedly rational adversaries and utilizes available data to learn the behavior models of the adversaries using the subjective utility quantal response (SUQR) model in order to improve the way the defender allocates its patrolling resources.

MIDAS has been successfully deployed and evaluated by the USCG in the Gulf of Mexico. Historical data on fish stock densities, USCG air and surface patrols, as well as IUU sightings and interdictions was used to construct the game model. Between July and September 2014, six aircraft patrols were generated weekly to protect a 80 by 60 nautical mile area on the US–Mexico border off the coast of Texas. This region represents a critical fishery for red snapper, a species that is highly lucrative to fish, and as such observes a high volume of IUU fishing. This evaluation period in the Gulf of Mexico represents the most sophisticated real-world deployment of security games to date. MIDAS is currently under review by the USCG and is being considered for further deployment in the Gulf of Mexico as well as in other fisheries nationwide.

16.4 Emerging Real-World Security Applications

16.4.1 Networked Domains

Beyond the deployed applications above, there are a number of emerging application areas. One such area of great importance is securing urban city networks, transportation networks, computer networks, and other network centric security domains. For example, after the terrorist attacks in Mumbai of 2008 [8], the Mumbai police have started setting up vehicular checkpoints on roads. We can model the problem faced by the Mumbai police as a security game between the Mumbai police and an attacker. In this urban security game, the pure strategies of the defender correspond to allocations of resources to edges in the network—for example, an allocation of police checkpoints to roads in the city. The pure strategies of the attacker correspond to paths from any *source* node to any *target* node—for example, a path from a landing spot on the coast to the airport. The strategy space of the defender grows exponentially with the number of available resources, whereas the strategy space of the attacker grows exponentially with the size of the network. In addressing this computational challenge, novel algorithms based on incremental strategy generation have been able to generate randomized defender strategies that scale up to the entire road network of Mumbai [19].

The Stackelberg game framework can also be applied to adversarial domains that exhibit "contagious" actions for each player. For example, word-of-mouth advertising/viral marketing has been widely studied by marketers trying to understand why one product or video goes "viral" while others go unnoticed. Counter-insurgency is the contest for the support of the local leaders in an armed conflict and can include a variety of operations such as providing security and giving medical supplies. These efforts carry a social effect beyond the action taken that can cause advantageous ripples through the neighboring population. Moreover, multiple intelligent parties attempt to leverage the same social network to spread their message, necessitating an adversary-aware approach to strategy generation. Game-theoretic approaches can be used to generate resource-allocations strategies for such large-scale, real-world networks [41, 42]. This interaction can be modeled as a graph with one player attempting to spread influence while another player attempts to stop the probabilistic propagation of that influence by spreading their own influence. This "blocking" problem models situations faced by governments/peacekeepers combatting the spread of terrorist radicalism and armed conflict with daily/weekly/monthly visits with local leaders to provide support and discuss grievances [15].

Game-theoretic methods are also appropriate for modeling resource allocation in cyber-security such as packet selection and inspection for detecting potential threats in large computer networks. The problem of attacks on computer systems and corporate computer networks gets more pressing each year. A number of intrusion detection and monitoring systems are being developed, e.g., deep packet inspection method that periodically selects a subset of packets in a computer network for analysis. The attacking/protecting problem can be formulated as a game between two players: the attacker (or the intruder) and the defender (the detection system). The actions of the attacker can be seen as sending malicious packets from a controlled computer to vulnerable computers. The objective of the defender is to prevent the intruder from succeeding by selecting the packets for inspection and subsequently thwarting the attack. However, packet inspections cause unwanted latency and hence the defender has to decide where and how to inspect network traffic. The computational challenge is efficiently computing the optimal defending strategies for such network scenarios [43].

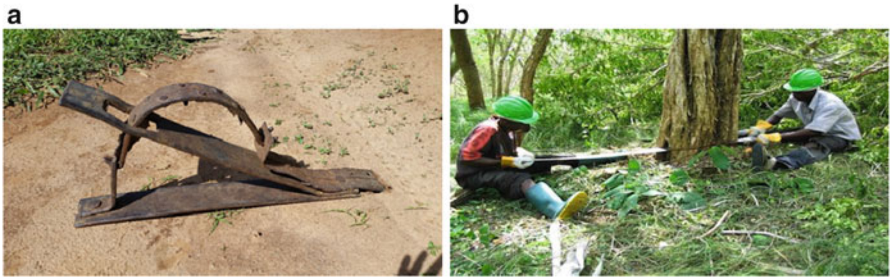


Fig. 16.5 Examples of illegal activities in green security domains. (**a**) An illegal trapping tool. (**b**) Illegally cutting trees.

16.4.2 Green Security Domains

A number of our newer applications are focused on resource conservation through suppression of environmental crime. One area is protecting forests [22], where we must protect a continuous forest area from extractors by patrols through the forest that seek to deter such extraction activity (Fig. 16.5). With limited resources for performing such patrols, a patrol strategy will seek to distribute the patrols throughout the forest, in space and time, in order to minimize the resulting amount of extraction that occurs or maximize the degree of forest protection. This problem can be formulated as a Stackelberg game and the focus is on computing optimal allocations of patrol density [22].

Endangered species poaching is reaching critical levels as the populations of these species plummet to unsustainable numbers. The global tiger population, for example, has dropped over 95 % from the start of the 1900s and has resulted in three out of nine species extinctions. Depending on the area and animals poached, motivations for poaching range from profit to sustenance, with the former being more common when profitable species such as tigers, elephants, and rhinos are the targets. To counter poaching efforts and to rebuild the species' populations, countries have set up protected wildlife reserves and conservation agencies tasked with defending these large reserves. Because of the size of the reserves and the common lack of law enforcement resources, conservation agencies are at a significant disadvantage when it comes to deterring and capturing poachers. Agencies use patrolling as a primary method of securing the park. Due to their limited resources, however, patrol managers must carefully create patrols that account for many different variables (e.g., limited patrol units to send out, multiple locations that poachers can attack at varying distances to the outpost). Our proposed system Protection Assistant for Wildlife Security (PAWS) aims to assist conservation agencies in their critical role of patrol creation by predicting where poachers will attack and optimizing patrol routes to cover those areas.

16.5 Scale Up to Real-World Problem Sizes

The wide use of Stackelberg games has inspired theoretical and algorithmic progress leading to the development of fielded applications, as described in Sect. 16.3. For example, DOBSS [35], an algorithm for solving Bayesian Stackelberg games, is central to the fielded application ARMOR in use at the Los Angeles International Airport [17]. Conitzer and Sandholm [9] gave complexity results and algorithms for computing optimal commitment strategies in Bayesian Stackelberg games, including both pure- and mixed-strategy commitments.

These early works assumed that the set of pure strategies for the players are given explicitly. Many real-world problems, like the FAMS and urban road networks, present billions of pure strategies to both the defender and the attacker. Such large problem instances cannot even be represented in modern computers, let alone solved using previous techniques. We have proposed models and algorithms that compute optimal defender strategies for massive real-world security domains [16, 18].

16.5.1 Scale Up with Defender Pure Strategies

In this section, we describe one particular algorithm ASPEN, that computes SSE in domains with a *very large* number of pure strategies (up to billions of actions) for the defender [16]. ASPEN builds on the insight that in many real-world game-theoretic problems, there exist solutions with *small support sizes*, which are mixed strategies in which only a small set of pure strategies are played with positive probability [28]. ASPEN exploits this by using a *strategy generation* approach for the defender, in which defender pure strategies are iteratively generated and added to the optimization formulation.

As an example, let us consider the problem faced by the FAMS. There are currently tens of thousands of commercial flights flying each day, and public estimates state that there are thousands of air marshals that are scheduled daily by the FAMS [24]. Air marshals must be scheduled on tours of flights that obey logistical constraints (e.g., the time required to board, fly, and disembark). An example of a schedule is an air marshal assigned to a round trip from Los Angeles to New York and back.

ASPEN [16] casts this problem as a security game, where the attacker can choose any of the flights to attack, and each air marshal can cover one schedule. Each schedule here is a feasible set of targets that can be covered together; for the FAMS, each schedule would represent a flight tour which satisfies all the logistical constraints that an air marshal could fly. A *joint schedule* then would assign every air marshal to a flight tour, and there could be exponentially many joint schedules in the domain. A pure strategy for the defender in this security game is a joint schedule. As mentioned previously, ASPEN employs strategy generation since all the defender pure strategies cannot be enumerated for such a massive problem. ASPEN decomposes the problem into a *master* problem and a *slave* problem, which

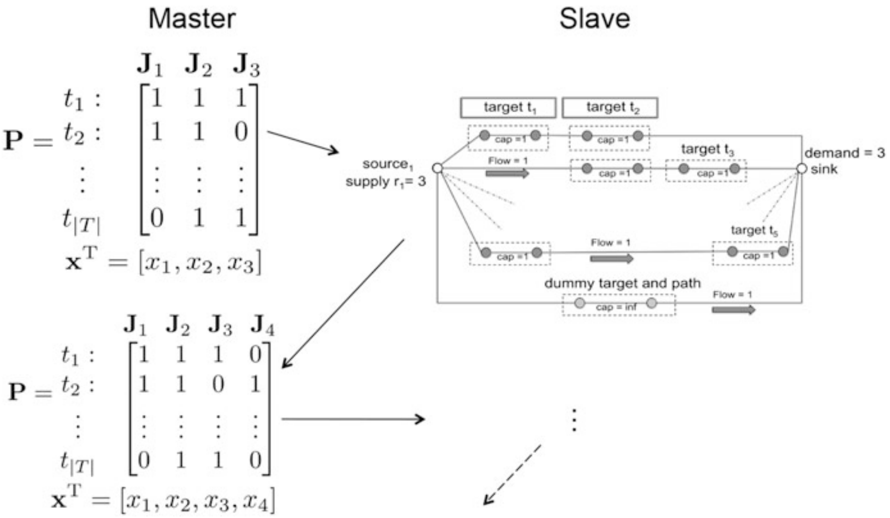


Fig. 16.6 Strategy generation employed in ASPEN: the schedules for a defender are generated iteratively. The *slave* problem is a novel minimum-cost integer flow formulation that computes the new pure strategy to be added to **P**; $\mathbf{J}_4$ is computed and added in this example

are then solved iteratively. Given a number of pure strategies, the master solves for the defender and the attacker optimization constraints, while the slave is used to generate a new pure strategy for the defender in every iteration.

The iterative process is graphically depicted in Fig. 16.6. The master operates on the pure strategies (joint schedules) generated thus far, which are represented using the matrix **P**. Each column of **P**, $\mathbf{J}_j$, is one pure strategy (or joint schedule). An entry P_{ij} in the matrix **P** is 1 if a target t_i is covered by joint-schedule $\mathbf{J}_j$, and 0 otherwise. The objective of the master problem is to compute **x**, the optimal mixed strategy of the defender over the pure strategies in **P**. The objective of the slave problem is to generate the best joint schedule to add to **P**. The best joint schedule is identified using the concept of *reduced costs*, which measures if a pure strategy can potentially increase the defender's expected utility (the details of the approach are provided in [16]). While a naïve approach would be to iterate over all possible pure strategies to identify the pure strategy with the maximum potential, ASPEN uses a novel minimum-cost integer flow problem to efficiently identify the best pure strategy to add. ASPEN always converges on the optimal mixed strategy for the defender.

Employing strategy generation for large optimization problems is not an "out-of-the-box" approach, the problem has to be formulated in a way that allows for domain properties to be exploited. The novel contribution of ASPEN is to provide a linear formulation for the master and a minimum-cost integer flow formulation for the slave, which enables the application of strategy generation techniques. Additionally, ASPEN also provides a branch-and-bound heuristic to reason over attacker actions. This branch-and-bound heuristic provides a further order of magnitude speed-up, allowing ASPEN to handle the massive sizes of real-world problems.

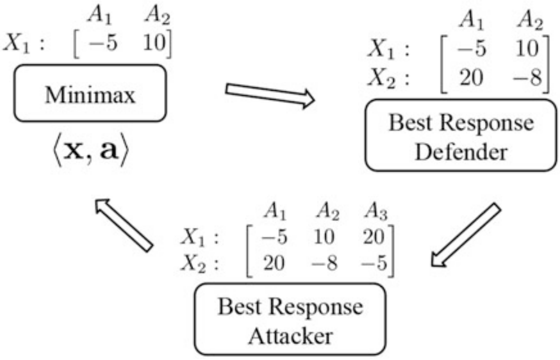


Fig. 16.7 Strategy generation employed in RUGGED: the pure strategies for both the defender and the attacker are generated iteratively

16.5.2 Scale Up with Defender and Attacker Pure Strategies

In domains such as the urban network security setting described in Sect. 16.4, the number of pure strategies of both the defender and the attacker are exponentially large. In this section, we describe the RUGGED algorithm [18], which generates pure strategies for both the defender and the attacker.

RUGGED models the domain as a zero-sum game, and computes the minimax equilibrium, since the minimax strategy is equivalent to the SSE in zero-sum games. Figure 16.7 shows the working of RUGGED: at each iteration, the minimax module generates the optimal mixed strategies $\langle \mathbf{x}, \mathbf{a} \rangle$ for the two players for the current payoff matrix, the Best Response Defender module generates a new strategy for the defender that is a best response against the attacker's current strategy **a**, and the Best Response Attacker module generates a new strategy for the attacker that is a best response against the defender's current strategy **x**. The rows X_i in the figure are the pure strategies for the defender, they would correspond to an allocation of checkpoints in the urban road network domain. Similarly, the columns A_j are the pure strategies for the attacker, they represent the attack paths in the urban road network domain. The values in the matrix represent the payoffs to the defender. The algorithm stops when neither of the generated best responses improve on the current minimax strategies.

The contribution of RUGGED is to provide the mixed-integer formulations for the best response modules which enable the application of such a strategy generation approach. RUGGED can compute the optimal solution for deploying up to 4 resources in real-city network with as many as 250 nodes within a reasonable time frame of 10 h (the complexity of this problem can be estimated by observing that both the best response problems are NP-hard themselves [18]). More recent work [19] builds on RUGGED and proposes SNARES, which allows scale-up to the entire city of Mumbai, with 10–15 checkpoints.

16.5.3 Scale Up with Mobile Resources and Moving Targets

In this section, we describe the CASS (Solver for Continuous Attacker Strategy) algorithm [10] for solving security problems where the defender has mobile patrollers to protect a set of mobile targets against the attacker who can attack these moving targets at any time during their movement. In these security problems, the sets of pure strategies for both the defender and attacker are continuous w.r.t the continuous spatial and time components of the problem domain. The CASS algorithm attempts to compute the optimal mixed strategy for the defender without discretizing the attacker's continuous strategy set; it exactly models this set using sub-interval analysis which exploits the piecewise-linear structure of the attacker's expected utility function. The insight of CASS is to compactly represent the defender's mixed strategies as a *marginal* probability distribution, overcoming the short-coming of an exponential number of pure strategies for the defender.

As a domain example, in the problem of protecting ferries described in Sect. 16.3.4, there are a number of ferries carrying hundreds of passengers in many waterside cities. These ferries are attractive targets for an attacker who can approach the ferries with a small boat packed with explosives at any time; this attacker's boat may only be detected when it comes close to the ferries. Small, fast, and well-armed patrol boats can provide protection to such ferries by detecting the attacker within a certain distance and stop him from attacking with the armed weapons. However, the numbers of patrol boats are often limited, thus the defender cannot protect the ferries at all times and locations.

CASS casts this problem as a *zero-sum* security game in which targets move along a *one-dimensional* domain, i.e., a straight line segment connecting two terminal points. This *one-dimensional* assumption is valid as in real-world domains such as ferry protection, ferries normally move back-and-forth in a straight line between two terminals (i.e., ports) around the world. Although the targets' locations vary w.r.t time changes, these targets have a fixed daily schedule, meaning that determining the locations of the targets at a certain time is straightforward. The defender has mobile patrollers (i.e., boats) that can move along between two terminals to protect the targets. While the defender is trying to protect the targets, the attacker will decide to attack a certain target at a certain time. The probability that the attacker successfully attacks depends on the positions of the patroller at that time. Specifically, each patroller possesses a protective circle of radius within which she can detect and try to intercept any attack, whereas she is incapable of detecting the attacker prior to that radius.

In CASS, the defender's strategy space is discretized and her mixed strategy is compactly represented using flow distributions. Figure 16.8 shows an example of a ferry transition graph in which each node of the graph indicates a particular pair of (location and time step) for the target. Here, there are three location points, namely A, B, and C on a straight line where B lies between A and C. Initially, the target is at one of these location points at the 5-min time step. Then the target moves to the next location point which is determined based on the connectivity between these

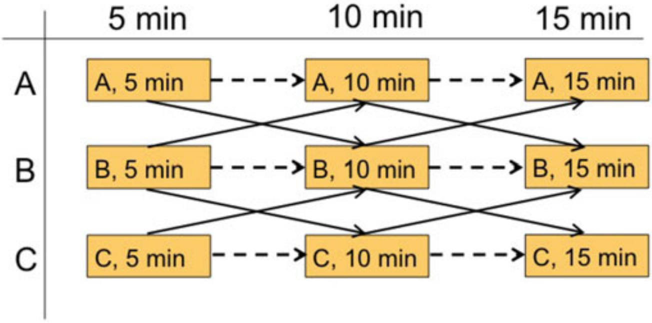


Fig. 16.8 An example of a ferry transition graph

points at the 10-min time step, and so on. For example, if the target is at the location point A at the 5-min time step, denoted by (A, 5 min) in the transition graph, it can move to the location point B or stay at location point A at the 10-min time step. The defender follows this transition graph to protect the target. A pure strategy for the defender is defined as a trajectory of this graph, e.g., the trajectory including (A, 5 min), (B, 10 min), and (C, 15 min) indicates a pure strategy for the defender. One key challenge of this representation for the defender's pure strategies is that the transition graph consists of an exponential number of trajectories, i.e., $O(N^T)$ where N is the number of location points and T is the number of time steps. To address this challenge, CASS proposes a compact representation of the defender's mixed strategy. Instead of directly computing a probability distribution over pure strategies for the defender, CASS attempts to compute the marginal probability that the defender will follow a certain edge of the transition graph, e.g., the probability of being at the node (A, 5 min) and moving to the node (B, 10 min). CASS shows that *any strategy in full representation can be mapped into a compact representation* as well as *compact representation does not lead to any loss in solution quality*. This compact representation allows CASS to reformulate the resource-allocation problem as computing the optimal *marginal* coverage of the defender over a number of $O(NT)$ the edges of the transition graph.

16.5.4 *Scale Up with Continuous Domains and Boundedly Rational Attacker*

As discussed in Sect. 16.3, natural resource conservation domains such as fishery protection introduce a unique set of challenges which must be addressed, namely *scalability* and *robustness*. For scalability, the defender is responsible for protecting a large patrol area and therefore must consider a large strategy space. Even if the patrol area is discretized into a grid or graph structure, the defender must still reason over an exponential number of patrol strategies. For robustness, the defender

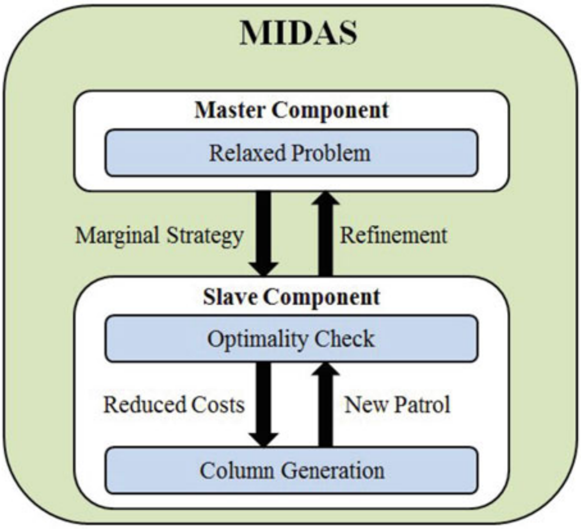


Fig. 16.9 Overview of the multiple iterative process within the MIDAS algorithm

must protect against *multiple* boundedly rational adversaries. Bounded rationality models, such as the quantal response (QR) model [31] and the SUQR model [32], introduce stochastic actions, relaxing the strong assumption in classical game theory that all players are perfectly rational and utility maximizing. These models are able to better predict the actions of human adversaries and thus lead the defender to choose strategies that perform better in practice. However, both QR and SUQR are non-linear models resulting in a computationally difficult optimization problem for the defender.

Previous work on boundedly rational adversaries has considered the challenges of scalability and robustness separately, in [47, 48] and [14, 49], respectively. The MIDAS algorithm was introduced to merge these two research threads for the first time by addressing scalability and robustness simultaneously. Figure 16.9 provides a visual overview of how MIDAS operates as an iterative process. Given the sheer complexity of the game being solved, the problem is decomposed using a master–slave formulation. The master utilizes multiple simplifications to create a relaxed version of the original problem which is more efficient to solve. First, a piecewise-linear approximation of the security game is taken to make the optimization problem both linear and convex. This is a modified version of the approach in [47], replacing the QR model of the adversary with SUQR and considering a robust maximin formulation over a set of boundedly rational adversaries. Second, the complex spatiotemporal constraints associated with patrols are initially ignored and then incrementally added back using cut generation.

Due to the relaxations, solving the master produces a marginal strategy **x** which is a probability distribution over targets. However, the defender ultimately needs a probability distribution over patrols. Additionally, since not all of the spatiotemporal constraints are considered in the master, the relaxed solution **x** may not be a feasible solution to the original problem. Therefore, the slave checks if the marginal strategy

x can expressed as a linear combination, i.e., probability distribution, of patrols by computing a one-norm minimization. If the one-norm distance is zero, the marginal distribution can be translated to a feasible pure strategy distribution which is in fact the optimal solution to the original problem. Otherwise, the marginal distribution is infeasible for the original problem. However, given the exponential number of patrol strategies, even performing this optimality check is intractable. Thus, column generation is used *within* the slave where only a small set of patrols is considered initially in the optimality check and the set is expanded over time. Much like previous examples of column generation in security games, e.g., [16], new patrols are added by solving a minimum-cost network flow problem using reduced cost information from the optimality check. If the optimality check fails, then the slave generates a cut which is returned to refine and constrain the master, incrementally bringing it closer to the original problem. The entire process is repeated until an optimal solution is found.

16.6 Address Uncertainty in Real-World Problems

Addressing uncertainty is a key challenge of solving real-world security problems. Traditional SSGs often assume that the defender has perfect information about the game payoff matrix as well as the attacker's behaviors. Moreover, she is supposed to be capable of exactly executing her patrolling strategy. However, due to limited data, the defender cannot precisely estimate such aspects, i.e., the payoff matrix or attacker's behaviors. Also, there is no guarantee that the defender can exactly follow the patrolling schedule as a result of unseen events that could change her patrolling strategy. These types of uncertainty could deteriorate the effectiveness of the defender's strategy and thus it is important for the defender to address them when generating strategy. This section of the book chapter describes several game-theoretic solutions to deal with uncertainty in SSGs.

16.6.1 Security Patrolling with Dynamic Execution Uncertainty

In security problems such as fare inspections in the Los Angeles Metro Rail system as described in Sect. 16.3.5, the targets, e.g., trains normally follow predetermined schedules, thus timing is an important aspect which determines the effectiveness of the defender's patrolling schedules (the defender needs to be at the right location at a specific time in order to protect these moving targets). However, as a result of execution uncertainty (e.g., emergencies or errors), the defender could not carry out her planned patrolling schedule in later time steps. For example, in real-world trials for TRUSTS carried out by Los Angeles Sheriff's Department (LASD), there is interruption (due to writing citations, felony arrests, and handling emergencies) in a significant fraction of the executions, causing the officers to miss the train they are supposed to catch as following the pre-generated patrolling schedule.

In this section, we present the Bayesian Stackelberg game model for security patrolling with dynamic execution uncertainty introduced by Jiang et al. [20] in which the uncertainty is represented using MDPs. The key advantage of this game-theoretic model is that patrol schedules which are computed based on Stackelberg equilibrium have contingency plans to deal with interruptions and are robust against execution uncertainty. Specifically, the security problem with execution uncertainty is represented as a two-player Bayesian Stackelberg game between the defender and the attacker. The defender has multiple patrol units while there are also multiple types of attackers which are unknown to the defender. A (naive) patrol schedule consists of a set of sequenced commands in the following form: at time t, the patrol unit should be at location l, and execute patrol action a. This patrol action a will take the unit to the next location and time if successfully executed. However, due to execution uncertainty, the patrol unit may end up at a different location and time. Figure 16.10 shows an example of execution uncertainty in a transition graph where if the patrol unit is currently at location A at the 5-min time step, she is supposed to take the on-train action to move to location B in the next time step. However, unlike CASS for ferry protection in which the defender's action is deterministic, there is a 10 % chance that she will still stay at location A due to execution uncertainty. These interactions of the defender with the environment when executing patrol can be represented as an MDP.

A key challenge of computing the SSE for this type of security problem is that the dimension of the space of mixed strategies for the defender is exponential in the number of states in terms of the defender's times and locations. Nevertheless, in many domains, the utilities have additional *separable* structure that the defender can exploit to efficiently compute an SSE of patrolling games with execution uncertainty. Specifically, when there exist *unit* utilities such as that both players' utilities can be represented as a linear combination of these *unit* utilities, the defender's Markov strategy can be obtained based on the marginal probabilities of

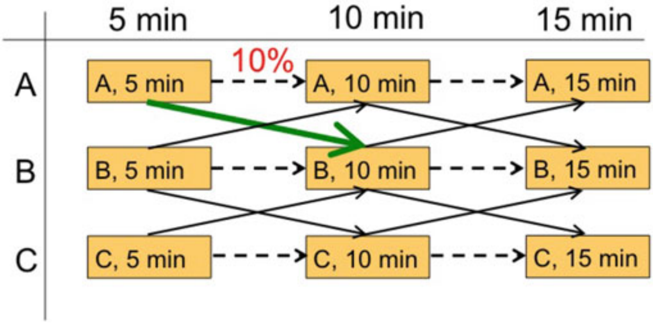


Fig. 16.10 An example of execution uncertainty in a transition graph

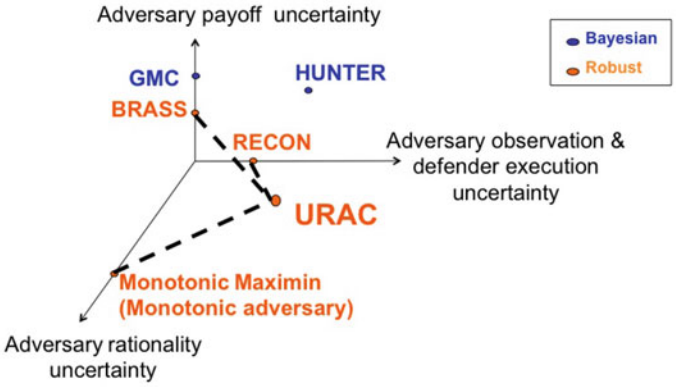

Adversary payoff uncertainty
Bayesian
Robust
GMC
HUNTER
BRASS
RECON
Adversary observation & defender execution uncertainty
URAC
Monotonic Maximin (Monotonic adversary)
Adversary rationality uncertainty

Fig. 16.11 Uncertainty space and algorithms

each patrolling unit reaching a state $s = (t, l)$, and taking action a. Here, the *unit* utilities only depend on a certain patrolling unit's state and action and a certain type of attacker. Therefore, instead of directly computing the mixed strategy, the defender attempts to compute the marginal probabilities which have dimensions polynomial in the sizes of the MDPs (the details of this approach are provided in [20]).

16.6.2 Security Patrolling with Unified Uncertainty Space

In this section, we present the two leading approaches for addressing uncertainty in security games in which the timing is not taken into account (which is different from the MDP-based approach described in the previous section). We first summarize the major types of uncertainties in security games as a three-dimensional uncertainty space with the following three dimensions (Fig. 16.11): (1) uncertainty in the adversary's payoff; (2) uncertainty related to the defender's strategy (including uncertainty in the defender's execution and the attacker's observation); and (3) uncertainty in the adversary's rationality. These dimensions refer to three key attributes which affect both players' utilities. The origin of the uncertainty space corresponds to the case with no uncertainty. Figure 16.11 also shows existing algorithms for addressing uncertainty in SSGs which follow the two main approaches: (1) modeling uncertainties based on Bayesian Stackelberg game models and (2) applying robust-optimization techniques. For example, BRASS [36] is a robust algorithm that only addresses attacker-payoff uncertainty while URAC (Unified Robust Algorithmic framework for addressing unCertainties) [33] is a unified robust algorithm that handles all types of uncertainty. In addition, HUNTER (Handling UNcerTainty Efficiently using Relaxation) [52] is a Bayesian-based algorithm that addresses all types of uncertainty except for the attacker-rationality uncertainty. While the Bayesian-based approach assumes a known distribution of uncertainties beforehand, the robust approach does not assume such prior knowledge.

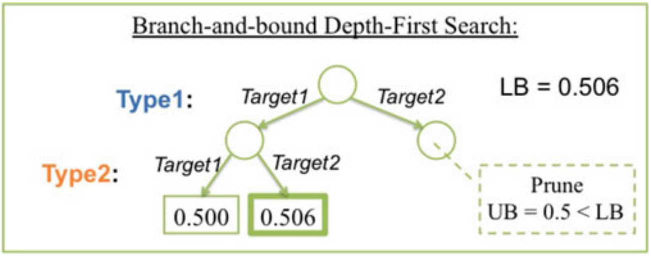


Fig. 16.12 Branch-and-bound depth first search

In the following, we will describe the two algorithms which are representatives of these two approaches: HUNTER (based on the Bayesian-based approach) and URAC (based on the robust approach).

16.6.2.1 Bayesian Approach

Overall, HUNTER is a novel algorithm for solving Bayesian Stackelberg games that can be used together with sample average approximation technique to solve Stackelberg games with uncertainty in the defender's execution and the attacker's observation [52]. Specifically, HUNTER attempts to compute the optimal mixed strategy for the defender against multiple attacker types with a prior distribution over the types. By exploiting the fact that the attacker is a perfectly rational player who will attack the optimal target with highest utility, HUNTER applies a best-first search for efficiently pruning the search tree that results from assigning attacker types to pure strategies as shown in Fig. 16.12. In other words, HUNTER first constructs the search tree by iteratively searching through all attacker types and all corresponding pure strategies for that attacker type. At each leaf node, the linear program at that node provides an optimal strategy for the defender such that the attacker's best response for every attacker type is the chosen target at that leaf node. Moreover, at internal nodes of the search tree (which corresponds to a partial assignment in which responses of a subset of attacker types are fixed), upper bounds and lower bounds of the optimal SSG solution are computed, which are then used to prune the search tree. As the size of the search tree is exponential in the number of targets and number of attacker types, finding tight upper bounds and lower bounds at internal nodes are essential in order to efficiently prune the search tree.

The key idea of HUNTER is to provide a tractable linear relaxation of Bayesian Stackelberg games that provides an upper bound efficiently at each of HUNTER's internal nodes based on finding a convex hull of all feasible solutions of the corresponding linear program at internal nodes. Figure 16.13 illustrates an example of constructing a convex hull of feasible solution regions of a two-target Bayesian security game with two attacker types. In Fig. 16.13a, each square corresponds to a partial assignment of an attacker type to a pure strategy, i.e., attacked target. The set

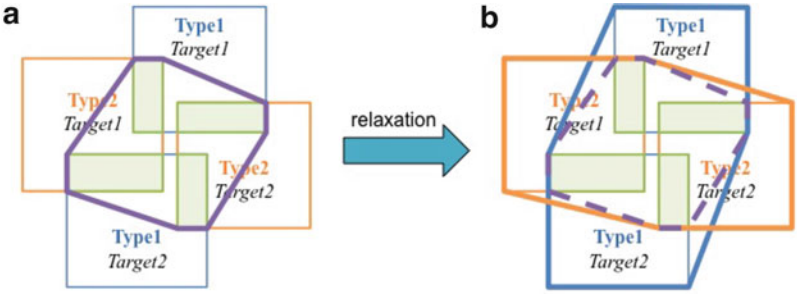


Fig. 16.13 An example of constructing a convex hull of feasible solution regions of a two-target Bayesian security games with two attacker types. HUNTER

of feasible solutions for the defender is the union of four disjoint green regions. As the optimal solution of a linear program is an extreme point of its feasible region, the linear program w.r.t the green regions is equivalent to a linear program with the same objective but w.r.t the convex hull of these four regions. However, the number of the disjoint regions is exponential in the number of targets and number of attacker types, finding a convex hull for these regions is computational. Therefore, HUNTER derives the relaxation of a Bayesian Stackelberg game by considering simpler convex hulls (of a small number of disjoint sets) (the blue and yellow regions shown in Fig. 16.13b) of which intersection is a super set of the convex hull of green regions. By solving this relaxation problem, HUNTER obtains an upper bound for the optimal solution of the Bayesian security game.

16.6.2.2 Robust Approach

In this section, we present the robust URAC algorithm for addressing a combination of all uncertainty types [33]. Consider an SSG where there is uncertainty in the attacker's payoff, the defender's strategy (including the defender's execution and the attacker's observation), and the attacker's behavior, URAC represents all these uncertainty types (except for the attacker's behaviors) using uncertainty interval. Instead of knowing exactly values of these game attributes, the defender only has prior information w.r.t the upper bounds and lower bounds of these attributes. For example, the attacker's reward if successfully attacking a target t is known to lie within the interval $[1, 3]$. Furthermore, URAC assumes the attacker monotonically responds to the defender's strategy. In other words, the higher the expected utility of a target, the more likely that the attacker will attack that target; however, the precise attacking probability is unknown for the defender. This monotonicity assumption is motivated by the Quantal Response model—a well-known human behavioral model for capturing the attacker's decision making [31].

Based on these uncertainty assumptions, URAC attempts to compute the optimal strategy for the defender by maximizing her utility against the worst-case scenario of uncertainty. The key challenge of this optimization problem is that it involves several types of uncertainty, resulting in multiple minimization steps for determining the worst-case scenario. Nevertheless, URAC introduces a unified representation of all these uncertainty types as a uncertainty set of attacker's responses. Intuitively, despite of any type of uncertainty mentioned above, what finally affects the defender's utility is the attacker's response, which is unknown to the defender due to uncertainty. As a result, URAC can represent the robust-optimization problem as a single maximin problem.

However, the infinite uncertainty set of the attacker's responses depends on the planned mixed strategy for the defender, making this maximin problem difficult to solve if directly applying the traditional method (i.e., taking the dual maximization of the inner minimization of maximin and merging it with the outer maximization—maximin now can be represented a single maximization problem). Therefore, URAC proposes a divide-and-conquer method in which the defender's strategy set is divided into subsets such that the uncertainty set of the attacker's responses is the same for every defender strategy within each subset. This division leads to multiple sub-maximin problems which can be solved by using the traditional method. The optimal solution of the original maximin problem is now can be computed as a maximum over all the sub-maximin problems.

16.7 Current Research

In this section we highlight several areas that we are actively doing research on, and point out some of the open research challenges.

16.7.1 Scalability

Driven by the growing complexity of applications, a sequence of algorithms for solving security games have been developed including DOBSS [35], ERASER [25], ASPEN [16], and RUGGED [18]. However, existing algorithms still cannot scale up to very large-scale domains. While RUGGED/SNARES computes optimal solutions much faster than any of the previous approaches, much work remains to be done for it to be applicable to complex heterogenous settings on large networks.

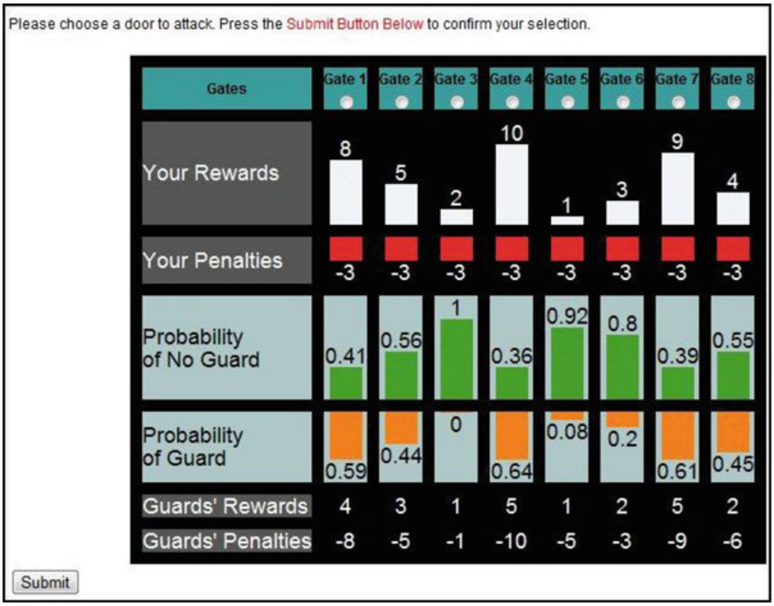
Besides strategy generation, another approach for dealing with an exponential number of pure strategies is to compactly represent mixed strategies as marginal probabilities of coverage on each of the targets. Because of the utility structure of security games, such marginal probabilities are sufficient to express the expected utility of the defender. Kiekintveld et al. [25] used this approach in ERASER to formulate the problem of computing SSE as a compact mixed-integer linear program. However, this approach is unable to deal with complex constraints on the defender resources [26]. Nevertheless, we have recently been able to use this approach for certain patrolling domains, including fare-enforcement patrols in urban transit systems [51] and boat patrols for protecting ferries [10]. In these domains a pure strategy is a patrol of a certain time duration over a set of locations, and the number of such pure strategies grow exponentially in the time duration. We were able to compactly represent mixed strategies as fractional flows on the *transition graph*, in which vertices are time–location pairs and arcs represent possible actions. This allowed us to formulate the optimization problems compactly which led to improved scalability. An open problem is to find other types of security domains in which the strategy space can be compactly represented. Another is to develop a hybrid approach that combines marginals and strategy generation.

16.7.2 Robustness

Classical game theory solution concepts often make assumptions on the knowledge, rationality, and capability (e.g., perfect recall) of players. Unfortunately, these assumptions could be wrong in real-world scenarios. Algorithms for the defender's optimal strategy have been proposed to take into account various uncertainties faced in the domain, including payoff noise [52], execution/observation error [50], and uncertain capability [1]. However, previous works assumed that the attacker knows (or with a small noise) the defender's mixed strategy. Recently An et al. [2] proposed a formal framework to model the attacker's belief update process as he observes instantiations of the defender's mixed strategy. The resulting optimization problem for the defender is non-linear and scalable computation remains an open issue. Furthermore, maximin is one of the leading robust method which is widely applied for addressing uncertainty in security games, which is known to be overly conservative. Minimax regret—an alternative less conservative robust criteria has just been applied recently to address payoff uncertainty [34]. The resulting optimization problem for using minimax regret is non-linear non-convex in both the defender strategy and the attacker's payoff and is thus computationally difficult. Moreover, addressing a combination of uncertainty using minimax regret has not been solved.

16.7.3 Adversary Modeling

One required research direction is addressing bounded rationality of human adversaries. This is a fundamental problem that can affect the performance of our game-theoretic solutions, since algorithms based on the assumption of the perfectly rational adversary are not robust to deal with deviations of the adversary from the optimal response. Recently, there has been some research on applying ideas from behavioral game theory (e.g., prospect theory [23] and quantal response [30]) within security game algorithms. One line of approaches is based on the quantal response model to predict the behaviors of the human adversary, and then to



Please choose a door to attack. Press the Submit Button Below to confirm your selection.

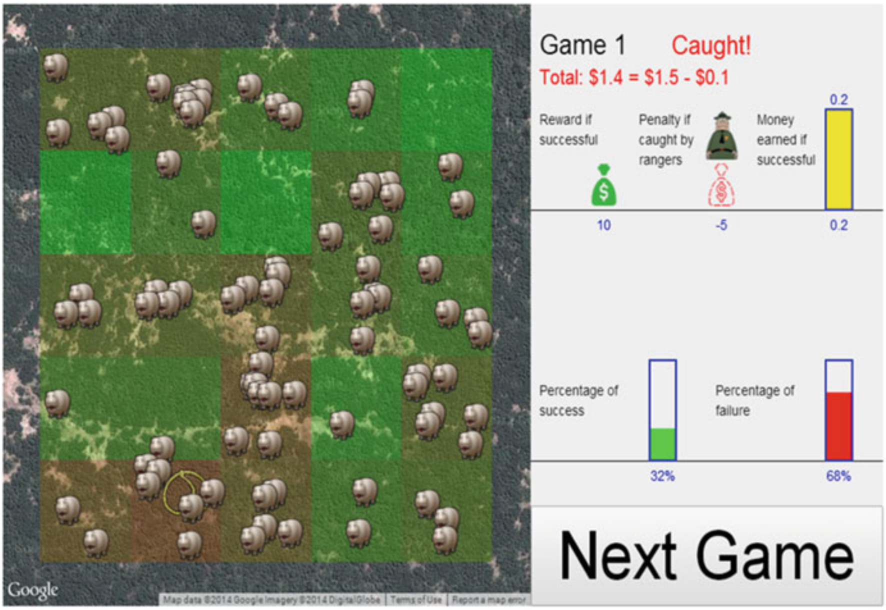
Gates	Gate 1 ○	Gate 2 ○	Gate 3 ○	Gate 4 ○	Gate 5 ○	Gate 6 ○	Gate 7 ○	Gate 8 ○
Your Rewards	8	5	2	10	1	3	9	4
Your Penalties	-3	-3	-3	-3	-3	-3	-3	-3
Probability of No Guard	0.41	0.56	1	0.36	0.92	0.8	0.39	0.55
Probability of Guard	0.59	0.44	0	0.64	0.08	0.2	0.61	0.45
Guards' Rewards	4	3	1	5	1	2	5	2
Guards' Penalties	-8	-5	-1	-10	-5	-3	-9	-6

Fig. 16.14 Interface of the Guards and Treasures game to simulate the LAX security scenario

compute optimal defender strategies against such behavior of the adversary. These include BRQR [46] which follows the logit quantal response (QR) [30] model, and subsequent work on SUQR models [32]. The parameters of these models are estimated by experimental tuning. Figure 16.14 shows the interface of an interactive game used in our human subject experiments, based on the security scenario at the LAX airport. The source code is available here. [1] Given the details for each target, the participants playing this game were asked to choose a target to attack. Data from a large set of participants on the Amazon Mechanical Turk (AMT) were collected and used to learn the parameters of the behavioral models to predict future attacks.

Experiments with the Guards and Treasures game were conducted only as a single-shot game where the adversary would observe the defender's strategy and then choose a target to attack and then the game would be over. While this may be true for domains like counter-terrorism, in other real-world domains like fisheries protection, or wildlife crime, there are repeated interactions between the defender and the adversary, where the game progresses in "rounds." We call this a Repeated SSG (RSSG) where in each round the defender would play a particular strategy and the adversary would observe that strategy and act accordingly. In order to simulate this scenario and conduct experiments to identify adversary behavior in such repeated settings, an online RSSG game was developed (shown in Fig. 16.15) and deployed.

[1] http://teamcore.usc.edu/projects/BGT/experiment.html.



Game 1 Caught!

Total: $1.4 = $1.5 - $0.1

Reward if successful: 10

Penalty if caught by rangers: -5

Money earned if successful: 0.2

Percentage of success: 32%

Percentage of failure: 68%

0.2
Next Game
Google
Map data ©2014 Google Imagery ©2014 DigitalGlobe | Terms of Use | Report a map error

Fig. 16.15 Interface of the Wildlife Poaching game to simulate an RSSG

In our game, human subjects play the role of poachers looking to place a snare to hunt a hippopotamus in a protected wildlife park. The portion of the park shown in the map is actually a Google Maps view of a portion of the QENP in Uganda. The region shown is divided into a 5 × 5 grid, i.e., 25 distinct cells. Overlaid on the Google Maps view of the park is a heat-map, which represents the rangers' mixed strategy x—a cell i with higher coverage probability x_i is shown more in red, while a cell with lower coverage probability is shown more in green. As the subjects play the game and click on a particular region on the map, they were given detailed information about the poacher's reward, penalty, and coverage probability at that region. However, the participants are unaware of the exact location of the rangers while playing the game, i.e., they do not know the pure strategy that will be played by the rangers, which is drawn randomly from mixed strategy x shown on the game interface. In our game, there were nine rangers protecting this park, with each ranger protecting one grid cell. Therefore, at any point in time, only 9 out of the 25 distinct regions in the park are protected. A player succeeds if he places a snare in a region which is not protected by a ranger, else he is unsuccessful. Similar to the Guards and Treasures game, here also we recruited human subjects on AMT and asked them to play this game repeatedly for a set of rounds with the defender strategy changing per round based on the behavioral model being used to learn the adversary's behavior.

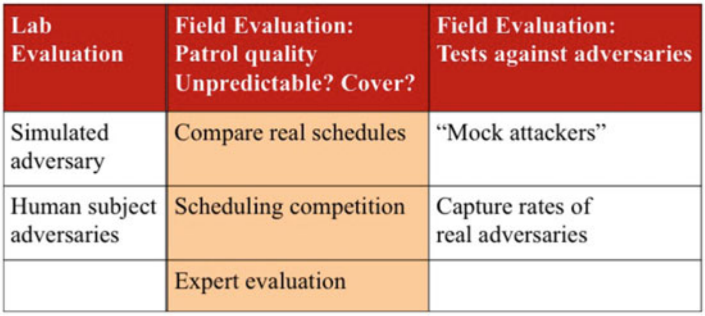
While behavioral models like (QR) [30] and SUQR [32] assume that there is a homogeneous population of adversaries, in the real-world we face heterogeneous populations of adversaries. Therefore Bayesian SUQR was proposed to learn the behavioral model for each attack [49]. PAWS is an application which was originally created using Bayesian SUQR. However, in real-world security domains, we may have very limited data, or may only have some limited information on the biases displayed by adversaries. An alternative approach is based on robust optimization: instead of assuming a particular model of human decision making, try to achieve good defender expected utility against a range of possible models. One instance of this approach is MATCH [37], which guarantees a bound for the loss of the defender to be within a constant factor of the adversary loss if the adversary responds non-optimally. Another robust solution concept is monotonic maximin [21], which tries to optimize defender utility against the worst-case monotonic adversary behavior, where monotonicity is the property that actions with higher expected utility is played with higher probability. Recently, there has been attempts to combine such robust-optimization approaches with available behavior data [14] for RSSGs. However, an open question of research is how these proposed models and algorithms will fare against human subjects in RSSGs. Furthermore, since real-world human attackers are sometimes distributed coalitions of socially, culturally, and cognitively biased agents, we may need significant interdisciplinary research to build in social, cultural, and coalitional biases into our adversary models.

16.7.4 Multi-Objective Optimization

In existing applications such as ARMOR, IRIS, and PROTECT, the defender is trying to maximize a single objective. However, there are domains where the defender has to consider multiple objectives simultaneously. Multi-objective security games (MOSGs) have been proposed to address the challenges of domains with multiple incomparable objectives [7]. In an MOSG, the threats posed by the attacker types are treated as different objective functions which are not aggregated, thus eliminating the need for a probability distribution over attacker types. Unlike Bayesian security games which have a single optimal solution, MOSGs have a set of Pareto-optimal (non-dominated) solutions which is referred to as the Pareto frontier. By presenting the Pareto frontier to the end-user, they may be able to better understand the structure of their problem as well as the trade-offs between different security strategies.

16.7.5 Evaluations: Lab Evaluation via Simulation and Field Evaluation

Evaluation in itself is a major challenge given the real-world deployment of these systems. It is difficult to define a baseline for the purpose of evaluation in security applications, as safety often trumps costs. Our evaluation focuses on presenting the benefit of our approach over prior approaches to security. We have conducted a



Lab Evaluation	Field Evaluation: Patrol quality Unpredictable? Cover?	Field Evaluation: Tests against adversaries
Simulated adversary	Compare real schedules	"Mock attackers"
Human subject adversaries	Scheduling competition	Capture rates of real adversaries
	Expert evaluation	

Fig. 16.16 Field evaluation

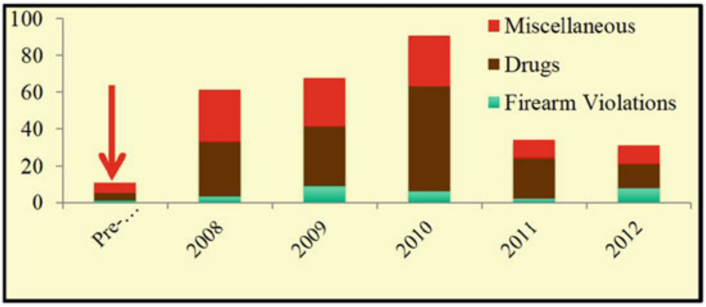


Fig. 16.17 ARMOR evaluation results

number of such evaluations: simulations, human subjects in the lab, assessment by domain experts internal and external to agencies deploying these applications, data from deployments (such as number of citations to fare-evaders), and adversary perspective teams (mock attacker teams) before and after deployment have all been used. We have already discussed simulations and human subject experiments in other parts of this chapter. Moreover, there are other evaluation approaches that we have tried, which are summarized in Fig. 16.16. In the following, we will discuss two of these approaches.

1. Data from deployment: data from the field, before and after deployment, supports our claim about improved security with our game-theoretic approach. Figure 16.17 shows the number of detected violations after ARMOR was deployed at LAX airport. As can be seen, the number of detected violations increased after our deployment and decreased in later years, suggesting better detection and deterrence effect of our approach. The patrol schedule for Boston port before and after our deployment of PROTECT (Fig. 16.18) clearly shows that there was a definite pattern in the patrols before PROTECT. In particular, there was low patrol for all targets on day 2, which could have been exploited by an attacker. In contrast, PROTECT provides almost the same level of patrol every day, with higher value targets patrol more often.

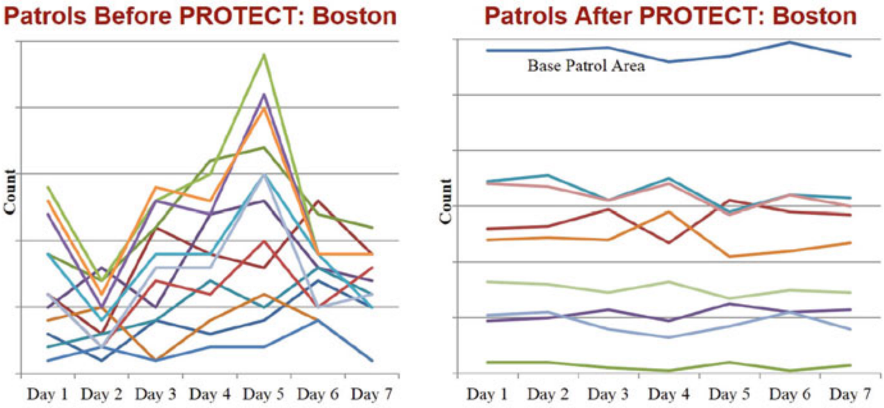


Fig. 16.18 PROTECT evaluation results: pre deployment (*left*) and post deployment patrols (*right*)

2. Mock attacker team: The USCG created an APT, a mock attacker team, to better understand the adversaries view of targets in the Boston port. This team, in addition to understanding the adversary's viewpoint, also gauged the effectiveness of patrol activities before and after deployment of PROTECT. The APT incorporates the adversary's known intent, capabilities, skills, commitment, resources, and cultural influences. In addition, it helps in identifying the level of deterrence projected at and perceived by the adversary. This analysis led to the conclusion that the effectiveness of deterrence increased from the before to after PROTECT deployment.

More detailed evaluations are discussed in the publications on the applications [11, 17, 39, 51], and more of these are discussed in [40].

16.8 Conclusion

Security is recognized as a world-wide challenge and game theory is an increasingly important paradigm for reasoning about complex security resource allocation. While the deployed game-theoretic applications have provided a promising start, very significant amount of research remains to be done. These are large-scale interdisciplinary research challenges that call upon multiagent researchers to work with researchers in other disciplines, be "on the ground" with domain experts, and examine real-world constraints and challenges that cannot be abstracted away.

References

1. An, B., Tambe, M., Ordonez, F., Shieh, E., Kiekintveld, C.: Refinement of strong Stackelberg equilibria in security games. In: Proceedings of the 25th Conference on Artificial Intelligence, pp. 587–593 (2011)
2. An, B., Kempe, D., Kiekintveld, C., Shieh, E., Singh, S., Tambe, M., Vorobeychik, Y.: Security games with limited surveillance. In: Conference on Artificial Intelligence (AAAI) (2012)
3. Avenhaus, R., von Stengel, B., Zamir, S.: Inspection games. In: Aumann, R.J., Hart, S. (eds.) Handbook of Game Theory, vol. 3, Chap. 51, pp. 1947–1987. North-Holland, Amsterdam (2002)
4. Breton, M., Alg, A., Haurie, A.: Sequential Stackelberg equilibria in two-person games. J. Optim. Theory Appl. **59**(1), 71–97 (1988)
5. Brown, G., Carlyle, M., Kline, J., Wood, K.: A two-sided optimization for theater ballistic missile defense. Oper. Res. **53**, 263–275 (2005)
6. Brown, G., Carlyle, M., Salmeron, J., Wood, K.: Defending critical infrastructure. Interfaces. **36**, 530–544 (2006)
7. Brown, M., An, B., Kiekintveld, C., Ordonez, F., Tambe, M.: Multi-objective optimization for security games. In: Proceedings of The 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2012)
8. Chandran, R., Beitchman, G.: Battle for Mumbai ends, death toll rises to 195. Times of India (29 November 2008). http://articles.timesofindia.indiatimes.com/2008-11-29/india/27930171_1_taj-hotel-three-terrorists-nariman-house
9. Conitzer, V., Sandholm, T.: Computing the optimal strategy to commit to. In: Proceedings of the ACM Conference on Electronic Commerce (ACM-EC), pp. 82–90 (2006)
10. Fang, F., Jiang, A., Tambe, M.: Optimal patrol strategy for protecting moving targets with multiple mobile resources. In: AAMAS (2013)
11. Fave, F.M.D., Brown, M., Zhang, C., Shieh, E., Jiang, A.X., Rosoff, H., Tambe, M., Sullivan, J.: Security games in the field: an initial study on a transit system(extended abstract). In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS) [Short paper] (2014)
12. Gatti, N.: Game theoretical insights in strategic patrolling: model and algorithm in normal-form. In: ECAI-08, pp. 403–407 (2008)
13. Hamilton, B.A.: Faregating Analysis. Report Commissioned by the LA Metro (2007). http://boardarchives.metro.net/Items/2007/11_November/20071115EMACItem27.pdf
14. Haskell, W.B., Kar, D., Fang, F., Tambe, M., Cheung, S., Denicola, L.E.: Robust protection of fisheries with compass. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence, pp. 2978–2983 (2014)
15. Howard, N.J.: Finding optimal strategies for influencing social networks in two player games. Master's thesis, MIT, Sloan School of Management (2011)

16. Jain, M., Kardes, E., Kiekintveld, C., Ordonez, F., Tambe, M.: Security games with arbitrary schedules: a branch and price approach. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp. 792–797 (2010)

17. Jain, M., Tsai, J., Pita, J., Kiekintveld, C., Rathi, S., Tambe, M., Ordonez, F.: Software assistants for randomized patrol planning for the LAX airport police and the federal air marshal service. Interfaces **40**, 267–290 (2010)

18. Jain, M., Korzhyk, D., Vanek, O., Pechoucek, M., Conitzer, V., Tambe, M.: A double oracle algorithm for zero-sum security games on graphs. In: Proceedings of the 10th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2011)

19. Jain, M., Tambe, M., Conitzer, V.: Security scheduling for real-world networks. In: AAMAS (2013)

20. Jiang, A., Yin, Z., Kraus, S., Zhang, C., Tambe, M.: Game-theoretic randomization for security patrolling with dynamic execution uncertainty. In: AAMAS (2013)

21. Jiang, A.X., Nguyen, T.H., Tambe, M., Procaccia, A.D.: Monotonic maximin: a robust Stackelberg solution against boundedly rational followers. In: Conference on Decision and Game Theory for Security (GameSec) (2013)

22. Johnson, M., Fang, F., Yang, R., Tambe, M., Albers, H.: Patrolling to maximize pristine forest area. In: Proceedings of the AAAI Spring Symposium on Game Theory for Security, Sustainability and Health (2012)

23. Kahneman, D., Tvesky, A.: Prospect theory: an analysis of decision under risk. Econometrica **47**(2), 263–291 (1979)

24. Keteyian, A.: TSA: Federal Air Marshals. http://www.cbsnews.com/stories/2010/02/01/earlyshow/main6162291.shtml (2010). Retrieved 1 Feb 2011

25. Kiekintveld, C., Jain, M., Tsai, J., Pita, J., Tambe, M., Ordonez, F.: Computing optimal randomized resource allocations for massive security games. In: Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 689–696 (2009)

26. Korzhyk, D., Conitzer, V., Parr, R.: Complexity of computing optimal Stackelberg strategies in security resource allocation games. In: Proceedings of the 24th AAAI Conference on Artificial Intelligence, pp. 805–810 (2010)

27. Leitmann, G.: On generalized Stackelberg strategies. J. Optim. Theory Appl. **26**(4), 637–643 (1978)

28. Lipton, R., Markakis, E., Mehta, A.: Playing large games using simple strategies. In: EC: Proceedings of the ACM Conference on Electronic Commerce, pp. 36–41. ACM, New York, NY (2003)

29. Lye, K., Wing, J.M.: Game strategies in network security. Int. J. Inf. Secur. **4**(1–2), 71–86 (2005)

30. McFadden, D.: Quantal choice analysis: a survey. Ann. Econ. Soc. Meas. **5**(4), 363–390 (1976)

31. McKelvey, R.D., Palfrey, T.R.: Quantal response equilibria for normal form games. Game Econ. Behav. **10**(1), 6–38 (1995)

32. Nguyen, T.H., Yang, R., Azaria, A., Kraus, S., Tambe, M.: Analyzing the effectiveness of adversary modeling in security games. In: Conference on Artificial Intelligence (AAAI) (2013)

33. Nguyen, T., Jiang, A., Tambe, M.: Stop the compartmentalization: unified robust algorithms for handling uncertainties in security games. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2014)

34. Nguyen, T.H., Yadav, A., An, B., Tambe, M., Boutilier, C.: Regret-based optimization and preference elicitation for Stackelberg security games with uncertainty. In: Proceedings of the National Conference on Artificial Intelligence (AAAI) (2014)

35. Paruchuri, P., Pearce, J.P., Marecki, J., Tambe, M., Ordonez, F., Kraus, S.: Playing games with security: an efficient exact algorithm for Bayesian Stackelberg games. In: Proceedings of the 7th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), pp. 895–902 (2008)
36. Pita, J., Jain, M., Ordóñez, F., Tambe, M., Kraus, S., Magori-Cohen, R.: Effective solutions for real-world Stackelberg games: when agents must deal with human uncertainties. In: The Eighth International Conference on Autonomous Agents and Multiagent Systems (2009)
37. Pita, J., John, R., Maheswaran, R., Tambe, M., Kraus, S.: A robust approach to addressing human adversaries in security games. In: European Conference on Artificial Intelligence (ECAI) (2012)
38. Arce, D.G., Sandler, T.: Terrorism and game theory. Simul. Gaming **34**(3), 319–337 (2003)
39. Shieh, E., An, B., Yang, R., Tambe, M., Baldwin, C., DiRenzo, J., Maule, B., Meyer, G.: PROTECT: a deployed game theoretic system to protect the ports of the United States. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2012)
40. Taylor, M., Kiekintveld, C., Tambe, M.: Evaluating deployed decision-support systems for security: challenges, analysis, and approaches. In: Tambe, M. (ed.) Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned. Cambridge University Press, Cambridge (2011)
41. Tsai, J., Nguyen, T.H., Tambe, M.: Security games for controlling contagion. In: Conference on Artificial Intelligence (AAAI) (2012)
42. Tsai, J., Qian, Y., Vorobeychik, Y., Kiekintveld, C., Tambe, M.: Bayesian security games for controlling contagion. In: Proceedings of the ASE/IEEE International Conference on Social Computing(SocialCom) (2013)
43. Vanek, O., Yin, Z., Jain, M., Bosansky, B., Tambe, M., Pechoucek, M.: Game-theoretic resource allocation for malicious packet detection in computer networks. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2012)
44. von Stackelberg, H.: Marktform und Gleichgewicht. Springer, Vienna (1934)
45. von Stengel, B., Zamir, S.: Leadership with commitment to mixed strategies. Technical Report, LSE-CDAM-2004-01, CDAM Research Report (2004)
46. Yang, R., Kiekintveld, C., Ordonez, F., Tambe, M., John, R.: Improving resource allocation strategy against human adversaries in security games. In: IJCAI (2011)
47. Yang, R., Ordonez, F., Tambe, M.: Computing optimal strategy against quantal response in security games. In: Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems, vol. 2, pp. 847–854 (2012)
48. Yang, R., Jiang, A.X., Tambe, M., Ordóñez, F.: Scaling-up security games with boundedly rational adversaries: a cutting-plane approach. In: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, pp. 404–410. AAAI Press, Menlo Park (2013)
49. Yang, R., Ford, B., Tambe, M., Lemieux, A.: Adaptive resource allocation for wildlife protection against illegal poachers. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2014)
50. Yin, Z., Jain, M., Tambe, M., Ordonez, F.: Risk-averse strategies for security games with execution and observational uncertainty. In: Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI), pp. 758–763 (2011)
51. Yin, Z., Jiang, A., Johnson, M., Tambe, M., Kiekintveld, C., Leyton-Brown, K., Sandholm, T., Sullivan, J.: TRUSTS: scheduling randomized patrols for fare inspection in transit systems. In: Proceedings of the 24th Conference on Innovative Applications of Artificial Intelligence (IAAI) (2012)

52. Yin, Z., Tambe, M.: A unified method for handling discrete and continuous uncertainty in Bayesian Stackelberg games. In: International Conference on Autonomous Agents and Multiagent Systems (AAMAS) (2012)

53. Zhang, C., Jiang, A.X., Short, M.B., Brantingham, P.J., Tambe, M.: Modeling crime diffusion and crime suppression on transportation networks: an initial report. In: SNSC 2013: The AAAI Fall Symposium 2013 on Social Networks and Social Contagion (2013)