1

Introduction and Overview of Security Games

Milind Tambe and Manish Jain

1.1 Introduction

Game theory's popularity continues to increase in a whole variety of disciplines, including economics, biology, political science, computer science, electrical engineering, business, law, and public policy. In the arena of security, where game theory has always been popular, there now seems to be an exponential increase in interest. This increase is in part due to the new set of problems our societies face, from terrorism to drugs to crime. These problems are ubiquitous. Yet, limited security resources cannot be everywhere all the time, raising a crucial question of how to best utilize them.

Game theory provides a sound mathematical approach for deploying limited security resources to maximize their effectiveness. While the connection between game theory and security has been studied for the last several decades, there has been a fundamental shift in the relationship due to the emergence of computational game theory. More specifically, with the development of new computational approaches to game theory over the past two decades, very large-scale problems can be cast in game-theoretic contexts, thus providing us computational tools to address problems of security allocations.

My research group has been at the forefront of this effort to apply computational game theory techniques to security problems. We have led a wide range of actual deployed applications of game theory for security. Our first application, *Assistant for Randomized Monitoring Over Routes* (ARMOR), successfully deployed game-theoretic algorithms at the Los Angeles International Airport (LAX) in 2007 and has been in use there ever since. In particular, ARMOR uses game theory to randomize allocation of police checkpoints and canine units. Our second application, *Intelligent Randomization in Scheduling* (IRIS), has been used by the U.S. Federal Air Marshal Service since 2009 to deploy air marshals on U.S. air carriers. A third application, *Game-theoretic Unpredictable and*

Randomly Deployed Security (GUARDS), for the U.S. Transportation Security Administration is being evaluated for a national deployment across more than 400 U.S. airports. A fourth application, *Port Resilience Operational/Tactical Enforcement to Combat Terrorism* (PROTECT), for the United States Coast Guard, is under development and has been demonstrated at the Port of Boston for evaluation; and many other agencies around the globe are now looking to deploy these techniques.

This set of applications and associated algorithms has added to the already significant interest in game theory for security. Yet this research is not confined to computer science; there has always been a wide variety of interest in game theory for security in researchers involved in risk, operations research, psychology, and other disciplines. Our applications of game theory have now generated interest in this topic from analysts and practitioners – police, security officials – who wish to deploy these solutions.

This book addresses some of this interest. My aim here is to bring together my research group's work over the past several years comprehensively in one book, describing the applications we have developed, the underlying research, and security officials' perspective on the problems. The book is designed to be of interest to (i) researchers and graduate students in the area of game theory for security who wish to understand the topic in more depth; (ii) security analysts and practitioners interested in obtaining an overview of this research (even if they skip details of our algorithms); and (iii) other researchers, generally familiar with game theory, who wish to jump into this area of research.

The book is divided into four parts. Part I is based on contributions of security officials; it provides their perspective on the challenges and needs for a game-theoretic approach to security. The remaining three parts contain papers I have co-authored with my current and former students, post-doctoral researchers, and colleagues. Part II provides an overview of applications we have developed, using key papers describing our applications. Part III will discuss our algorithms in depth using selected papers and finally, Part IV will outline some key directions of future research. To those familiar with game theory, and particularly computer scientists, all four parts will be easily accessible. *Those unfamiliar with game theory can still follow the first two parts.*

The rest of this chapter provides a high-level and *informal* overview of the material presented in the rest of the book. We begin in Section 1.2 by briefly outlining the key motivation for applying game theory to security; of course, Part I of this book delves much more deeply into this motivation. More importantly, Section 1.2 will also provides relevant background in game theory and, in particular, the types of games used in our work. Next, Section 1.3 provides

an overview of Part II of this book, that is, of the deployed applications. Section 1.4 similarly provides an overview of Part III; and Section 1.5, of Part IV.

1.2 Motivation: Security Games

Part I provides us with the motivation for the security work discussed in this book. A key motivating concern is infrastructure security: We have to protect our ports, airports, buses and trains, transportation, and other infrastructure. Yet we often have limited security resources to accomplish this goal, which means we cannot provide a security cover for everything twenty four hours a day. Security resources have to be deployed selectively. Unfortunately, our adversaries can monitor our defenses and exploit any patterns in these selective deployments. For example, if we check trains only on Tuesdays and Thursdays, an adversary will observe and exploit this pattern. Similarly, in patrolling an airport, if the patrols are at Terminal 1 at 9 AM, Terminal 2 at 10 AM, Terminal 3 at 11 AM, an adversary will learn this information. The key here is that an adversary conducts surveillance and then plans an attack exploiting any patterns in our security activities. Chapter 2 by Erroll Southers in Part I of the book provides a detailed outline of the terrorist planning cycle and the role of surveillance.

Game theory can provide us with a method to allocate limited security resources to protect infrastructure, taking into account the different weights of different targets and an adversary's response to any particular infrastructure protection strategy. Typically, the solution suggested by using a game-theoretic approach is a weighted randomization strategy. Security resources are allocated in a randomized fashion, but with higher weights on some targets than others, as specified by a game-theoretic solution concept. To accomplish this goal, we rely in particular on specific types of games called Bayesian Stackelberg games. For the benefit of those who are unfamiliar with these games, or perhaps even with game theory in general, we will provide a brief, informal introduction. Obviously, this is a very short introduction to a topic that has entire textbooks devoted to it (Fudenberg and Tirole, 1991); knowledgeable readers may skip one or both subsections as appropriate.

1.2.1 Game Theory

Game theory is an abstract mathematical theory for analyzing interactions among multiple intelligent actors, where the actors may be people, corporations, nations, intelligent software agents, or robots. In a security context, the intelligent actors may be security forces or police, on the one hand, and

adversaries on the other. In providing a mathematical basis for understanding intelligent actors' interactions with each other, game-theoretic approaches assume that these intelligent actors will anticipate each other's moves, and act appropriately.

The origins of game theory are in the 1940s with the work of John von Neumann and Oskar Morgenstern (Neumann and Morgenstern, 1944), although some readers may be more familiar with John Nash's celebrated work in the 1950s (Nash, 1951). While it started out in the area of economics, game theory has now been used in analysis in many academic disciplines: political science, philosophy, biology, and others. Perhaps the latest entry into this arena of applying and making contributions in game theory is the discipline of computer science. This has led to computational approaches to game theory, thus providing us computational tools to analyze large-scale interactions of multiple intelligent actors. We have leveraged precisely these computational techniques in our work.

1.2.2 Bayesian Stackelberg Games

In our work, we appeal to a special class of games, called Bayesian Stackelberg games. Before we get into Bayesian Stackelberg games, I will attempt to explain the notion of Stackelberg games (so named due to their origins in the work of Heinrich von Stackelberg [Stackelberg, 1934]). I will explain this class of games starting with a simple example, but before doing so, I emphasize again our assumption that we have limited security resources, which must protect multiple potential infrastructure targets of varying importance.

Consider a simple airport with two terminals, Terminals 1 and 2. There is only one police unit to protect the terminals and one adversary. Terminal 1 happens to be more important than Terminal 2 in this example. The game in Figure 1.1 shows this situation; by a "game" we mean a mathematical description of the problem of interaction between the multiple actors. The result is the matrix shown below, with the police's choice of actions depicted along the rows and the adversary's choice of actions shown along the columns. In this case, the police can protect Terminal 1 or Terminal 2; the adversary can attack Terminal 1 or Terminal 2. The numbers in the matrix describe the payoffs to the police and the adversary, as described in Figure 1.1.

Knowing that Terminal 1 is more important than Terminal 2, the police may choose to always protect Terminal 1. However, an intelligent adversary will conduct surveillance and, after learning that the police always protect Terminal 1, will attack Terminal 2. That is, the police have played the strategy described by the Terminal 1 row in the game matrix; and the adversary has responded with

		Adversary	
		Terminal 1	Terminal 2
Defender	Terminal 1	5, –3	–1, 1
	Terminal 2	–5, 5	2, –1

Figure 1.1. Stackelberg game.

the strategy described by the Terminal 2 column. We assume here that since there are no police at Terminal 2, the adversary's attack succeeds. The entry (–1,1) at the intersection of the intersection of the Terminal 1 row and the Terminal 2 column describe the payoffs to the police and the adversary. Specifically, the police will get a payoff of –1 since the adversary's attack succeeds, and the adversary gets a payoff of 1. In this case, we are assuming all payoffs are in the range of –5 to 5. The payoff is a way of quantitatively representing the loss or gain due to a successful attack. For example, it may specifically represent some measure of loss of life or economic loss or a combination of both and other factors.

This payoff is each actor's (police or adversary) view of his/her own utility. It is thus quite possible that the loss to the adversary may not be symmetric with the gain to the police and vice versa. For example, had the adversary attacked Terminal 1 when the police were stationed at Terminal 1, the police would have captured the adversary; then the adversary would be the one with a negative payoff of –3, and the police, having captured the adversary, will have a positive payoff of 5. The reason the adversary's payoff may not be –5 is that the adversary may view even a failed attack as not the worst outcome, possibly due to the publicity received for the attempt to attack an important terminal.

How can we arrive at a precise estimate of such a payoff in a game? Typically, these payoffs result via knowledge acquisition from domain experts. In our own applications, these payoffs arise from calculations based on a set of answers to a set of key questions (created by domain experts) about the impact of adversary success and failure quantified in terms of loss of lives, damage to property, and other measures; in some cases, these payoffs are generated by other researchers with expertise in risk analysis. Later, we will also discuss algorithms that handle uncertainty over such payoffs. For now, we assume that these payoffs are specified with precision. We will return to the payoffs after the discussion of Bayesian Stackelberg games.

Of course, an intelligent adversary will not attack Terminal 1 if the police always guard Terminal 1. Similarly, if the police were to switch their strategy and always protect Terminal 2, an adversary conducting surveillance will observe

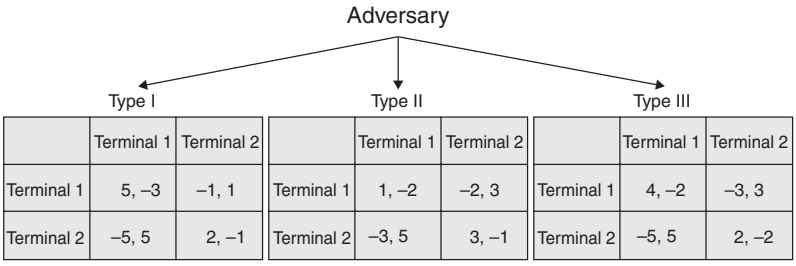
that, and subsequently will attack Terminal 1. In this case, the adversary again gets a positive reward of 5 and the police get a negative reward of –5. Thus, an adversary can easily defeat any deterministic police strategy of choosing to always protect either Terminal 1 or Terminal 2.

If, however, the police were to randomize their actions, for example, if they were to be at Terminal 1 60% of the days, and spend the remaining 40% of the days at Terminal 2, then that would lead to a better result. An adversary conducting surveillance will know that the police spend 60% of the days at Terminal 1, and 40% at Terminal 2, but precisely where they will be tomorrow remains unknown. This increases adversary uncertainty and improves the expected reward for the police.

These types of games are called Stackelberg games because the police commit first to a strategy, for example, the 60%/40% splitting of their resources between Terminal 1 and Terminal 2. An adversary acts after conducting surveillance. Notice that the police have committed to a randomized strategy, also called a "mixed strategy." The adversary responds with a single action – an attack – not a randomized response; the adversary's reaction here is a "pure strategy" reaction (in this simple game, we did not model the adversary's action of switching to another airport entirely; had we done so, that would be another pure strategy reaction modeling the adversary's being deterred from attacking this airport). Thus, the model matches the attack methodology provided in Part I of this book: Adversaries conduct surveillance over an extended period of time to get an understanding of police (security) strategy and then launch an attack on a target. The assumption here is that the adversary will only know the general resource allocation strategy (e.g., its 60%/40% distribution of resources) due to prior surveillance but will not know exactly how the security resources will be allocated on the day of the planned attack (because the schedule for the day is generated at random).

These Stackelberg games are also called "attacker-defender games," and we will sometimes use the terms "defender" and "attacker" playing this game. A key point to note here is that we assume that the attacker (adversary) has perfect knowledge of the defender's mixed strategy, and that the adversary will react rationally to this strategy, maximizing his/her own expected utility. (In the rest of this chapter, to disambiguate the defender and attacker in our descriptions, we will use "she" to denote the defender and "he" to denote the attacker.)

The key question of course is whether the 60%/40% splitting of resources is the optimal way to divide the defender's resources. Or should it be 65%/35%, or 50%/50%? We focus on this question of optimal division of resources. With two terminals and one police unit, we could easily solve this problem by hand. With



	Terminal 1	Terminal 2
Terminal 1	5, –3	–1, 1
Terminal 2	–5, 5	2, –1

	Terminal 1	Terminal 2
Terminal 1	1, –2	–2, 3
Terminal 2	–3, 5	3, –1

	Terminal 1	Terminal 2
Terminal 1	4, –2	–3, 3
Terminal 2	–5, 5	2, –2

Figure 1.2. Bayesian Stackelberg game.

hundreds of targets and multiple police units, the problem requires efficient computational solution approaches.

Furthermore, the problem in reality is even more complex: From Stackelberg games, we now move into Bayesian Stackelberg games. In Bayesian Stackelberg games, we admit uncertainty over different adversary types. For example, one adversary type may consider Terminal 1 to be more important than Terminal 2. Another adversary type may consider Terminal 2 to be equal in importance to Terminal 1 for some symbolic reason. A third adversary type may not be able to attack Terminal 1 effectively, and so on. Thus, there is not only one payoff matrix, but many of them, each corresponding to a different adversary type, as shown in Figure 1.2.

1.2.3 Security Games

In our work, we often appeal to a further specialization of the Bayesian Stackelberg games called "security games." Security games have the characteristic that what is good for the attacker is bad for the defender and vice versa. However, we do not require that the sum of the payoffs be zero. If the sum was always zero, then we would have zero-sum games. However, in general, the games we address need not be zero-sum. There has been a significant discussion in the literature on why these games are not zero sum (Powell 2007), but some reasons could be that the adversary views some targets as particularly important for his/her audience for their symbolic value, whereas they may not be of equal importance to the police. Or as mentioned earlier, an adversary may not view even a failed attack as a negative outcome because of the publicity and fear it generates. Or the adversary may need to incur a significant cost in mounting a particular attack that may not be particularly important to the police.

In essence, in a security game, if an attacker attacks a target that was covered (protected) by the defender, then the attacker has a worse payoff than if the attacker had attacked the same target when it was not covered. For example,

when the attacker in Figure 1 attacked Terminal 1, when the police were protecting Terminal 1, the attacker has a worse payoff (payoff of –3) than when the attacker attacked Terminal 1 when it was not covered/protected by the police (payoff of 5). This situation is reversed for the defender.

Our more recent work has begun to extend this notion of security games, so that a target is not merely covered or uncovered. Rather, there may be a probability associated with how well a target is covered because of a particular security action; and the attacker may have multiple options for attacking the target as well. Appropriate generalization of this concept remains an issue for active research.

Given such Bayesian Stackelberg games, whether in the form of security games or not, the key is to find the optimal allocation of security resources that will optimize the defender's expected reward. Technically, what we are interested in finding is a *strong Stackelberg equilibrium* (SSE). Formal technical definitions of SSE are provided in the papers in Part III of this book. However, the key to remember in SSE is that it assumes that the adversary has perfect knowledge of the defender's mixed strategy and reacts with perfect rationality to that strategy, choosing to react in a way that maximizes his expected utility. In a SSE, the defender has no incentive to change her strategy since it is the optimal strategy, and the attacker has no incentive to change his response because it is the optimal response to the defender's mixed strategy.

Deterrence from attacking the set of targets being protected can be modeled in such games by introducing a new action for the attacker: The *Not Attack Targets* action (which may actually involve attacking another target or performing another action that provides a certain positive rewards). In some cases, given the defender's SSE strategy, the attacker's best response is to not attack any of the targets that the defender is aiming to protect; that is, the targets have been hardened enough that the attacker is deterred from attacking this set of targets. The key point here is that deterrence emerges due to the adversary's choice of this new action; this action is chosen by the adversary only if it is the adversary's best response to the defender's mixed strategy. However, beyond this initial step, understanding and modeling deterrence in more depth remains a topic for future work.

As mentioned earlier, for a small 2 x 2 game as shown earlier, we might be able to compute the SSE by hand. When we have hundreds of targets and even just ten resources, the problem of computing SSE becomes extremely difficulty to solve by hand and requires a computational solution. Even this computational approach runs into difficulties as we scale up beyond that – because it is difficult to enumerate in memory all of the defender's possible choices. Our contributions are in finding an optimal solution quickly.

1.3 Overview of Part II: Applications of Security Games

Part II of this book discusses our applications in depth; key papers include discriptions of one ARMOR (Pita et al., 2008), IRIS (Tsai et al., 2009), and GUARDS (Pita et al., 2011). In addition to providing a brief overview of these applications, this section describes some on-going work not reported in Part II, and some opportunities for further applications.

1.3.1 ARMOR

Our first application of security games was ARMOR (Assistant for Randomized Monitoring Over Routes). As detailed in Part I, this application emerged in 2007 after police at LAX approached us with the question of how to randomize deployment of their limited security resources. For example, they have six inbound roads into LAX, and they wished to set up checkpoints. There are not enough police to have checkpoints on all roads at all times. So the question is where and when to set up these checkpoints. Similarly, they have eight terminals but not enough explosive-detecting canine units to patrol all terminals at all times of the day (a canine unit is limited by the number of hours a dog can work per day). Given that LAX may be under surveillance by adversaries, the question is where and when to have the canine units patrol the different terminals.

The police approached us in April 2007, after we had designed our first set of algorithms. Although the algorithms were ready, we needed to spend several months acquiring knowledge, learning how different police units performed their duties, what constraints there were in terms of shifts of operations, obtaining detailed data on passenger loads at different times of day at different terminals, and so on. The passenger data, for example, influences how payoffs are determined in our underlying game representation – our adversaries would want to cause maximum harm to civilians and the higher the passenger load, the higher the payoff to the adversaries.

By August 2007, after multiple iterations, the police started using ARMOR in setting up checkpoints and, later for canine patrols. The backbone of ARMOR is the algorithms for solving Bayesian Stackelberg games; they recommend a randomized pattern for setting up checkpoints and canine unit patrols. Police provide inputs like the number of available canine units; ARMOR then provides to the police an hour-by-hour schedule of where to set up canine patrols.

ARMOR continues to be used at LAX and has undergone periodic updates to its software. The ARMOR system has received numerous accolades. I discuss some criteria for evaluation of ARMOR a little later in this chapter, and Chapter 13 is dedicated to evaluation of all of our deployed systems.

1.3.2 IRIS

After our ARMOR experience, we were fortunate enough to be contacted by the Federal Air Marshals Service (FAMS). Their challenge is to randomize allocations of air marshals to flights to avoid predictability by adversaries conducting surveillance (e.g., these might be part of an insider threat), yet to provide adequate protection to more important flights. We are focused in particular on some sectors of international flights. Even within that domain, there are a very large number of flights over a month, and not enough air marshals to cover all of them.

To accomplish the goal of randomizing the allocation of air marshals to flights, we constructed a system called IRIS (Intelligent Randomization in Scheduling). We delivered the system to FAMS in the Spring of 2009. After extensive testing, they started using this system in October 2009. At its backend, IRIS casts the problem it solves as a Stackelberg game and, in particular, as a security game. We focused on the special nature of the security game framework to build fast algorithms for IRIS. Initially, IRIS used the ERASER-C algorithm as described in (Tsai et al., 2009); more recently, IRIS switched to the ASPEN algorithm (Jain et al., 2010). Both ERASER-C and ASPEN are discussed in Part III of this book.

1.3.3 GUARDS

After IRIS, our next focus was GUARDS (Game-theoretic Unpredictable and Randomly Deployed Security). GUARDS was developed in collaboration with the United States Transportation Security Administration (TSA) to assist in resource allocation tasks for airport protection at more than 400 U.S. airports. Unlike ARMOR and IRIS, which focus on one installation/application and one security activity (e.g., canine patrol or checkpoints) per application, GUARDS reasons with multiple security activities, diverse potential threats, and also hundreds of end users. The goal for GUARDS is to allocate TSA personnel to security activities that protect the airport infrastructure; GUARDS does not check passengers.

GUARDS again utilizes a Stackelberg game but generalizes beyond security games and develops a novel solution algorithm for these games. GUARDS has been delivered to TSA and is currently undergoing evaluation and testing for scheduling practices at an undisclosed airport. If successful, TSA intends to incorporate the system into its unpredictable scheduling practices nationwide.

1.3.4 Beyond ARMOR/IRIS/GUARDS

Beyond ARMOR, IRIS, and GUARDS, we have recently started a pilot project with the United States Coast Guard to build a new system called PROTECT