Randomizing Security Activities with Attacker Circumvention Strategies

James Pita
University of Southern California
Los Angeles, CA 90089

Chris Kiekintveld
University of Southern California
Los Angeles, CA 90089

Milind Tambe
University of Southern California
Los Angeles, CA 90089

Michael Scott
University of Southern California
Los Angeles, CA 90089

ABSTRACT

Game theoretic methods for making resource allocation decision in security domains have attracted growing attention from both researchers and security practitioners, including deployed applications at both the LAX airport and the Federal Air Marshals Service. We develop a new class of security games designed to model decisions faced by the Transportation Security Administration and other agencies in protecting airports, ports, and other critical infrastructure. Our model allows for a more diverse set of security activities for the defensive resources than previous work, which has generally focused on interchangeable resources that can only defend against possible attacks in one way. Here, we are concerned in particular with the possibility that adversaries can circumvent specific security activities if they are aware of common security measures. The model we propose takes this capability into account and generates more unpredictable, diverse security policies as a result—without resorting to an external value for entropy or randomness.

Solving these games is a significant computational challenge, and existing algorithms are not capable of solving realistic games. We introduce a new method that exploits common structure in these problems to reduce the size of the game representation and enable faster solution algorithm. These algorithms are able to scale to make larger games than existing solvers, as we show in our experimental results.

Categories and Subject Descriptors

I.0 [**Computing Methodologies**]: General

General Terms

Game Theory, Security

Keywords

Game Theory, Stackelberg, Security, Resource Allocation

1. INTRODUCTION

Security officials face many difficult decisions in how to provide security for critical infrastructure, high-profile events, and other potential targets of criminal or terrorist attacks. Game theory is increasingly viewed as a powerful tool for modeling these decisions, due in part to the ability of these models to account for adaptive adversaries and to identify optimal randomized strategies for security forces. This basic idea has been applied in several contexts, including autonomous robot patrolling [1, 3], scheduling checkpoints and canine patrols at the Los Angeles International Airport (LAX) [14], and scheduling Federal Air Marshals (FAMS) on flights [8]. The final two examples are real-world software systems that are deployed to make critical resource allocation decisions using game-theoretic reasoning.

Our work in this paper is motivated by the challenges of a different class of security allocations problems faced by agencies in charge of security at airports, ports, and other large physical areas. We develop a new class of game models that offer a richer model of the possible security strategies for the defender, allowing the specification of both the *area(s)* that are defended along with the *activity* that is executed by the security resource. An important aspect of this model is that it represents asymmetric knowledge between the attacker and defender. While we have a detailed understanding of the possible security policies, we have less detail about all of the possible attacker strategies—in reality, it is difficult if not impossible to predict all of the possible attack scenarios a sophisticated attacker might use. Instead, we introduce the possibility for attackers to *circumvent* specific security measures into the model, at some cost. As we show in our analysis, randomized policies are much more difficult for the attacker to plan around, increasing the value of unpredictable activities. Previous models have directly added values for "entropy" as part of the objective function; in our model, the value for randomizing among similar activities is driven by the circumvention capability of the adversary.

Another way that our model generalizes previous work is by allowing multiple resources to be assigned to the same physical area (or target), increasing the level of protection afforded to this area. In addition, the most general form of our model allows for different levels of effectiveness to be associated with different activities, so some are more likely than others to prevent attacks. This is used in particular to model the effects of executing activities such as patrols in physically adjacent areas. While the main protective effect is in the area directly being patrolled, there may be some visibility and capability to respond to incidents in nearby areas, providing a reduced level of protection for those areas as an additional benefit.

The additional richness of this model comes at a computational cost, and computing solutions to this model using existing algorithms is not feasible. In particular, the standard Stackelberg approach is capable of representing these games only by enumerating an exponential number of strategies for both the attacker and defender. While similar issues have been addressed in recent work on algorithms for the FAMS game [8], these methods cannot be directly applied here because they are not designed to handle cases where resources may carry out different activities and provide varying levels of protection. We develop a novel compact representation for this game based on identifying classes of strategies that can be treated symmetrically for the purposes of computing an optimal solution. By exploiting these symmetries we are able to solve much larger game instances than previous methods, which we demonstrate in our experimental results.

2. RELATED WORK

There is work on resource allocation for security settings that uses both game-theoretic approaches as well as more standard optimization frameworks. Our work focuses on developing more detailed models of the possible security measures than previous game-theoretic approaches. A particularly unique aspect of our model is the generic capability that attackers have to circumvent specific security measures.

There are three main areas of related work. The first apply optimization techniques to model the security domain, but do not address the strategic aspects of the problem. These methods provide a randomization strategy for the defender, but they do not take into account the fact that the adversaries can observe the defender's actions and then adjust their behavior. Examples of such approaches include [13, 15] which are based on learning, Markov Decision Processes (MDPs) and Partially Observable Markov Decision Processes(POMDPs). As part of this work, the authors model the patrolling problem with locations and varying incident rates in each of the locations and solve for optimal routes using a MDP framework. Another example is the "Hypercube Queueing Model" [9] which is based on queueing theory and depicts the detailed spatial operation of urban police departments and emergency medical services. Such frameworks can address many of the problems we raise, including different area values and increasing uncertainty by using many possible patrol routes. However, they fail to account for the possibility that an intelligent attacker will observe and exploit patterns in the security policy. If a policy is based on the historical frequency of attacks, it is essentially a reactive policy, one that an intelligent adversary can exploit.

A second set of work uses Stackelberg games to model a variety of security domains. Game-theoretic models have been applied in a variety of security settings, such as protecting critical infrastructure [6, 11, 14]. Lawrence [18] applies Stackelberg games in the context of screening visitors entering the US. They have also been used for studying missile defense systems [4] and for studying the development of an adversary's weapon system [5]. Other recent work is on randomized security patrolling using Stackelberg games for generic "police and robbers" scenarios [7] and perimeter patrols [1]. Our work differs from the previous work in that it allows for a more fine-grained representation of security domains. A representation that allows for different levels of protection from security measures and unique security activities that are no longer interchangeable. It also allows for adversary models that do not explicitly represent modes of attack, as in much of the previous work, but still manages to capture some of the adversary's capabilities.

The final set of related work is the application of game theoretic techniques that are not based on Stackelberg games to security applications. Security problems are increasingly studied using game-theoretic analysis, ranging from computer network security [17, 10] to terrorism [16]. Babu et al [2] have worked on modeling passenger security system at US airports using linear programming approaches, however, their objective is to classify the passengers in various groups and then screen them based on the group they belong to. Thus, although game theory has been used in security domains in the past, our work focuses on extending these domains to relax some of the previous assumptions that have been made.

3. MOTIVATING DOMAINS

Our work here is motivated by a large number of security domains where the challenge is to protect a large physical environment from attackers using limited security resources. Such domains include examples like railroad and subway systems, power generation facilities, shipping ports, and airports. In any of these domains there exist a wide variety of possible security measures that could be implemented to provide protection for the facility, including activities such as perimeter patrols, screening inbound vehicles, or verifying the credentials of employees or passengers. The organizations tasked with providing security for these domains include local law enforcement, port authorities, and the Transportation Security Administration (TSA). These organizations face the challenging problem of maximizing the protection of the critical infrastructure using a limited number of available resources.

Assigning resources is complicated by the fact that there are many different areas of a large facility where resources could be allocated. For example, in an airport there are public areas (e.g., ticketing and check in areas), boarding areas, as well as secured areas such as the aircraft runways. Some of these areas are physically distant, while others may be adjoining or accessible through other areas. In addition, these areas may have different values from a security perspective, since they have different numbers of people and some may have other important assets (e.g., aircraft or expensive machinery).

There is also a wide variety of different kinds of tasks or activities that security forces could perform. Each type of activity may be able to prevent different kinds of harmful actions or events. For example, a security activity might be screening baggage for harmful substances, patrolling the perimeter for unauthorized entrants, or verifying the identify of passengers. These activities may have different effects on protecting different areas of the facility. For example, if passengers are screened at the check-in area then this may also help protect the individual terminals since it can stop a potential threat from entering the terminal area through the check-in area. Although a security activity may protect more than one area at a time, the level of protection it provides to each area may vary. Returning to our check-in area example, although screening passengers at the check-in area may protect terminals from unwanted passengers, there may be other threats to the terminals like a worker who enters from a different area. Thus screening only helps partially protect the terminal area, and different combinations of security activities may provide additional protection against a broader range of threats. We will refer to the combination of a security activity and the area where it is performed as a security "operation."

The goal of an attacker is to find a successful strategy to attack some area of the facility. This decision depends both on the goals of the attacker and on the security measures taken by the security forces.

Areas that have less security will be less costly to attack, but they may also not be desirable depending on the overall objective of the attacker. To increase the chances of success an attacker may also try to specifically circumvent particular kinds of security measures that the attacker believes are likely to be in place. This could also take the form of choosing a particular type of attack vector that will not be detected or prevented by known security measures. However, this becomes increasingly difficult as the number and variety of security activities taking place in any given area increases.

Based on these domains we are interested in, we develop a game model to capture the salient features of these domains. Our base model is a Stackelberg game, similar to previous models discussed in Section 2. However, we extend these models to incorporate decisions about both location and the types of security activities being carried out, as well as to incorporate the possibility that attackers can circumvent specific security measures at some additional cost. We begin by introducing a basic version of the model, and then describe a compact representation of this model that allows for more scalable solution methods. We then present an extension to this model that allows greater flexibility in modeling different levels of protection for different types of security activities.

4. SECURITY GAMES WITH COMPLEX ACTIONS

We begin with a high-level description of our game model before giving a more formal definition. Our game model has two players, an attacker and a defender (i.e., the security forces). The defender is trying to prevent attacks on a large physical space—such as an airport or port facility—that can be partitioned into a number of smaller areas. To prevent attacks the defender is able to execute various actions (i.e., "operations") using security resources; these actions are associated with a specific area(s), and perform a particular kind of activity.

The attacker's goal is to successfully attack one of the areas in the facility, but to do so the attacker must also avoid any security activities being performed in the area. As the defender allocates more resources to protect one area, it becomes more difficult for the attacker to successfully attack this area. In our model, areas may have different characteristics, including the payoff each player receives if there is a successful attack on the area, or a failed attack attempt in the area. The defender's actions also have different effects on each area, providing different degrees of protection in different areas. It may also be more or less difficult for attackers to circumvent security measures in different areas.

Real-world terrorist attacks are based on careful planning and often use surveillance or other means to gather detailed information about security procedures. To model this, previous work has adopted Stackelberg game models where the defender moves first and commits to a (randomized) strategy for deploying security resources. The attacker is able to observe this strategy and plan the best possible attack, based on this knowledge. A standard solution concept for these game is a Strong Stackelberg Equilibrium (SSE) in which the defender chooses an optimal mixed strategy, assuming that the attacker will choose an optimal strategy in response. We adopt this Stackelberg framework and solution concepts for the model and algorithms presented in this paper.

In the remainder of this section we define the possible strategies for both the defender and the attacker, and then describe how payoffs are assigned for the possible outcomes of the game. We initially assume for expository purposes that each operation affects exactly one area, and that all operations are identical in how effective they are at preventing attacks. We relax both of these assumptions in Section 6.

4.1 Defender Strategies

We denote the defender by Θ, and the set of defender's pure strategies by $\sigma_\Theta \in \Sigma_\Theta$. In our model the defender is able to execute a variety of security activities called *operations*, which we denote by $O = \{o_1, \ldots, o_m\}$. Each individual operation has two components. The first is the type of activity that the operation represents, and the second is the area(s) where the activity is performed. For now, we assume that each operation affects exactly one area from the set of areas denoted by $A = \{a_1, \ldots, a_n\}$.

The defender has limited resources available for running defensive operations, and so is able to run a maximum of K operations on any day. An assignment of K resources to a set of K operations represents a single strategy $\sigma_\Theta \in \Sigma_\Theta$. For example, if there are three operations, $O = \{o_1, o_2, o_3\}$ and two resources available, one possible pure strategy for the defender is to assign these two resources to o_1 and o_3. The defender's mixed strategies $\delta_\Theta \in \Delta_\Theta$ are the possible probability distributions over Σ_Θ.

4.2 Attacker Strategies

The attacker is denoted by Ψ, and the set of pure strategies for the attacker is given by $\sigma_\Psi \in \Sigma_\Psi$. Similarly, the attacker's mixed strategies are probability distributions over the pure strategies and are denoted by $\delta_\Psi \in \Delta_\Psi$. Each pure strategy for the attacker corresponds to selecting a single area $a_i \in A$ to attack. In principle, the attacker will also choose a specific mode of attack. However, in security domains it is typically not feasible to enumerate all possible modes of attack, and attackers often develop new or modified versions of attacks that have not been seen before. This is particularly the case when security measures are known and predictable, so that attackers are able to specifically plan countermeasures to circumvent the security procedures.

Rather than try to enumerate specific attack scenarios and run the risk of failing to include important possibilities, we model the attacker's strategies at a higher level of abstraction. In addition to selecting an area to attack, the attacker chooses a subset of the possible operations that could be run in that area to avoid, or circumvent. Circumventing operations will increase the attacker's chances of success, but comes with a fixed cost that is a parameter of the model. This cost could capture a variety of different things, such as using more sophisticated technology or additional people to launch the attack, or switching to a less ideal means of attack that is less destructive. For example, if the defender is searching baggage for harmful substances (the operation), but not screening passengers, the attacker could choose to use a vest bomb as their mode of attack which would avoid the baggage screening. Formally, a pure strategy for the attacker consists of an area a_i and a subset of the operations in O to circumvent. It is only necessary for the attacker to circumvent operations that affect area a_i.

4.3 Payoff Definition

Payoffs for each player are defined over all possible joint pure-strategy outcomes: $\Omega_\Theta : \Sigma_\Psi \times \Sigma_\Theta \rightarrow \Re$ for the defender and similarly for the attacker. The payoff functions are extended to mixed strategies in the standard way by taking the expectation over pure-strategy outcomes. The first component of the payoff depends on which area the attacker chooses to attack, and whether

or not the attack was successful. We define four values for each area: $V_\Theta^d(a_i)$ and $V_\Theta^a(a_i)$ for the defender and $V_\Psi^d(a_i)$ and $V_\Psi^a(a_i)$ for the attacker. Here d signifies the area being successfully defended while a signifies the area being successfully attacked so $V_\Theta^d(a_i) > V_\Theta^a(a_i)$ for the defender while $V_\Psi^d(a_i) < V_\Psi^a(a_i)$ for the attacker.

The probability of success or failure depends on both the operations the defender is running in the attacked area, and the set of operations the attacker is circumventing. We define $\lambda(a_i, \sigma_\Theta)$ to be the set of operations $o_i \in \sigma_\Theta$ that affect area a_i (which might be the empty set). For now, we assume that an attack is successful if and only if $\lambda(a_i, \sigma_\Theta) \subseteq \sigma_\Psi$. This assumes that every operation has a 100% chance of preventing the attack unless it is circumvented by the attacker.

After the attack is determined to be successful or not the payoff also depends on which operations the attacker has chosen to circumvent. We introduce this cost as a function $C(a_i, \sigma_\Psi)$ which is the cost of circumventing the set of operations chosen in σ_Ψ for the attacked area. The larger the set of operations σ_Ψ contains the larger the cost becomes, so it is more difficult to successfully attack areas that are more heavily defended. We include the area because circumventing certain operations may be easier in some areas than in others depending on factors like layout, daily activities in that area, and the number of people who are regularly present in that area. This cost is deducted from the attacker's payoff and added to the defender's overall payoff, resulting in the following overall payoffs for both players in the case of a successful attack:

$$V_\Theta^a(a_i) + C(a_i, \sigma_\Psi) \quad (1)$$

$$V_\Psi^a(a_i) - C(a_i, \sigma_\Psi) \quad (2)$$

The payoff for a failed attack is identical except for substituting V_Θ^a with V_Θ^d for the defender and the same for the attacker. To further explain the game representation we have just outlined and how payoffs are calculated in this game we will turn to a concrete example. In this example there are two areas, $A = \{a_1, a_2\}$, and four operations $O = \{o_1, o_2, o_3, o_4\}$. Here o_1 and o_2 affect only a_1, and o_3 and o_4 affect only a_2. For the follower we set $V_\Psi^a(a_1) = 5$, and $V_\Psi^d(a_1) = -1$ for the first area and $V_\Psi^a(a_2) = 10$, and $V_\Psi^d(a_2) = -5$ for the second area. For the defender we set $V_\Theta^d(a_1) = 2$, and $V_\Theta^a(a_1) = -10$, for the first area and $V_\Theta^d(a_2) = 5$, and $V_\Theta^a(a_2) = -20$ for the second area. Finally we set the costs as $C(a_1, o_1) = C(a_1, o_2) = 2$ and $C(a_2, o_3) = C(a_2, o_4) = 3$. Figure 1 shows a physical representation of this game with corresponding payoffs. In our example there will be 2 resources to assign, $K = 2$. We show the possible outcomes of this game in normal-form in Table 1.

In Table 1 the first value represents the defender's payoff and the second value represents the attacker's payoff. The attacker's actions are represented first by the area selected for the attack and then by the operations avoided. For instance, the second column represents the attacker choosing area a_1 and avoiding operation o_1 where the third column represents the attacker choosing area a_1 and avoiding operation o_2. To illustrate how these values are translated into the table lets look at the case where the defender chooses o_1, o_3 and the attacker attacks area a_1 while avoiding o_1. Since the attacker avoided all the operations we were running in that area he succeeds in his attack, thus he receives $V_\Psi^a(a_1)$ or 5 points. However, the cost to avoid o_1, or $C(a_1, o_1)$, is 2 so the attacker only receives 3 points (5 - 2). On the defender's side we go through similar logic to arrive at a payoff of -8 (-10 + 2).

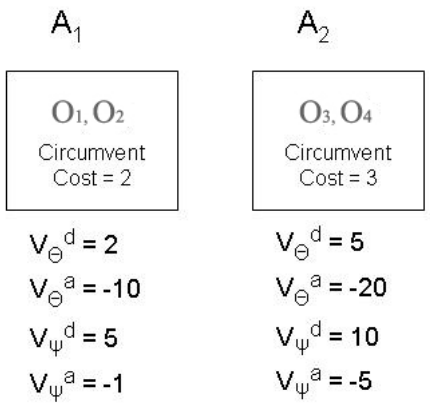

Figure 1: Game Example

	$a_1 : \emptyset$	$a_1 : o_1$	$a_1 : o_2$	$a_2 : \emptyset$	$a_2 : o_3$	$a_2 : o_4$
o_1, o_2	2, -1	4, -3	4, -3	-20, 10	-17, 7	-17, 7
o_1, o_3	2, -1	-8, 3	4, -3	5, -5	-17, 7	8, -8
o_1, o_4	2, -1	-8, 3	4, -3	5, -5	8, -8	-17, 7
o_2, o_3	2, -1	4, -3	-8, 3	5, -5	-17, 7	8, -8
o_2, o_4	2, -1	4, -3	-8, 3	5, -5	8, -8	-17, 7
o_3, o_4	-10, 5	-8, 3	-8, 3	5, -5	8, -8	8, -8

Table 1: Example payoffs for sample game

	$a_1 : \emptyset$	$a_1 : \gamma_1$	$a_2 : \emptyset$	$a_2 : \gamma_2$
γ_1, γ_1	2, -1	4, -3	-20, 10	-17, 7
γ_1, γ_2	2, -1	-2, 0	5, -5	-4.5, -5
γ_2, γ_2	-10, 5	-8, 3	5, -5	8, -8

Table 2: Example compact version

Given this setup we can construct a standard Stackelberg game. Namely, we have outlined the strategy space, Σ_Θ, for the defender to be the set of all possible combinations of K operations and the strategy space, Σ_Ψ, for the attacker to be the set of all possible circumvention strategies for each area. We have also outlined how payoffs are determined based on the strategy chosen by the defender and the attacker. Particularly, the attack fails or succeeds based on whether the attacker has circumvented the necessary operations at the area he chooses to attack and the cost of the attack is factored into both the defender's and attacker's payoffs irrespective of whether the attack succeeds or fails. Given the normal-form representation of any of these games similar to that shown in Table 1, this game can be solved using the fastest known general Stackelberg solver, DOBSS [12].

5. COMPACT REPRESENTATION

Although setting up our new problem as described is solvable using a general Stackelberg solver, it does not scale well as the size of the game increases. Both the attacker and defender strategy spaces grow combinatorially as the number of defender operations increases. We introduce a compact representation that exploits similarities in defender operations to reduce the number of strategies

that must be enumerated and considered when finding an optimal solution to the game.

5.1 Exploiting Identical Operations

First, we identify operations that provide coverage to the same areas, and have the same circumvention costs; so far, all operations within a given area are identical. Let $\gamma_i \in \Gamma$ represent the sets of operations that can be grouped together because they have identical properties. The key is that each of these operations will have the same effect on the payoffs, so we can reason about them as a group and only consider the *number* of operations of each type that are selected by the defender or circumvented by the adversary. We can show that in the optimal solution, the selection probabilities and circumvention strategies take a simple form. In particular, we now argue that it is optimal for the defender to distribute probability uniformly at random across all operations within a set γ_i, so that all operations are chosen with equal probability in the solution. Given this, we only need to know how many operations are selected from each set in order to compute the expected payoffs for each player in the optimal solution.

PROPOSITION 1. *Selecting each operation $o_j \in \gamma_i$ with equal probability provides the maximum expected payoff for the defender.*

Proof Sketch: Let the vector $X = \langle x_1, x_2, \ldots, x_j \rangle$ represent the total probability with which each operation associated with a given area is selected according to some mixed strategy δ. Without loss of generality, assume that this vector is sorted in descending order such that $x_1 \geq x_2 \geq \cdots \geq x_n$. The attacker strictly prefers to circumvent operations that are selected with higher probability, so the attacker will always choose to circumvent operations $x_1 \ldots x_m$ for any number of circumvented operations m. Now, consider the alternative defender strategy $\hat{\delta}$ with uniform coverage probabilities $\hat{x}_1 = \hat{x}_2 = \cdots = \hat{x}_n = (\sum_{i=1}^{n} x_i)/n$. For any m operations that the attacker could circumvent, $\sum_{i=m+1}^{n} x_i \leq \sum_{i=m+1}^{n} \hat{x}_i$ because the vectors have the same sum and we have eliminated the m maximum elements of X. Therefore, the attacker succeeds no more frequently against strategy δ than $\hat{\delta}$, and the defender's expected payoff is at least as great for the uniform strategy $\hat{\delta}$. ■

A strategy $\sigma_\Theta \in \Sigma_\Theta$ can now be represented by the number of resources assigned to each set of identical operations γ_i. For example, if there are two sets γ_1 and γ_2, and the defender has 2 available resources, the possible strategies are to assign both to γ_1, one to each set, or both to γ_2 (assuming at least two operations in each set). The original strategy space consists of all possible ways to select two operations from n possible operations, which is much larger than the compact strategy space as n grows large.

We now define define $\lambda(\gamma_i, \sigma_\Theta)$ to be number of resources assigned to γ_i in the strategy σ_Θ. We also use the notation Υ_{a_i} to represent the set of all γ_i that affect area a_i. Finally, we define Q^Θ to be the vector of resource assignments over Γ where Q_i^Θ is the number of resources assigned to γ_i.

Given that the defender strategy uniformly distributes resources among all operations $o_j \in \gamma_i$ we also know that it does not matter which specific operations the attacker chooses to circumvent from the set γ_i. For any given number of operations circumvented, the expected payoff is identical regardless of which specific operations within the set are chosen. Therefore, we can use a similar compact representation for the attacker strategy space as for the defender, reasoning only over the aggregate number of operations of each type rather than specific operations. Specifically, a strategy σ_Ψ is represented by which area the attacker chooses to attack and then by how many operations from each set γ_i the attacker circumvents. Similar to the defender, this is a much smaller strategy space than the original strategy space which enumerates all possible unique circumvention strategies. We define Q^Ψ to be the vector of the number of operations circumvented over Γ where Q_i^Ψ is the number of operations circumvented from the set γ_i.

A concrete example of this representation is presented in Table 2, for the same game shown in Table 1. In this representation there are only 3 pure strategies for the defender: assign both resources to γ_1 operations, assign one resource to γ_1 and one to γ_2, or assign both resources to γ_2. Similarly, for the attacker there are now only 2 circumvention options per area: circumvent no operations or circumvent one operation of the appropriate set γ_i. We will now explain how payoffs are calculated in this new compact version and how these payoffs map back to the full representation.

5.2 Computing Payoffs in the Compact Representation

We have defined a compact representation for both the defender and attacker strategies. It remains to describe how payoffs are calculated for combinations of these strategies, and how these payoffs reflect the payoffs in the original game. To compute the payoffs for a combination of strategies we must first calculated the probability that an attack succeeds. For any given defender strategy the defender resources allocated to each operation type (Q_i^Θ) are uniformly distributed over the operations in γ_i. In addition, the attacker will receive an identical payoff for any set of operations circumvented within γ_i. Therefore, we can select an arbitrary pure strategy from the full representation for the attacker which circumvents each fixed number of operations; we will refer to this strategy as τ.

We now describe how to compute the expected payoffs for both players for attacker strategy τ by computing the probability that the attacker will succeed against the defender strategy. Let $\xi_i \in \Xi_i$ represent the possible combinations of operations in $\gamma_i \in \Upsilon_{a_i}$, where a_i is the area attacked in τ. The attack succeeds if and only if the operations circumvented in τ are a superset of the operations in ξ. For each ξ we compute the number of times the attacker fails, f_i, by counting the occurrences where all operations in ξ are not circumvented in τ. The attacker succeeds in all other cases, denoted by w_i. The attacker's overall probability of failure taking into account all types of operations is given by $\epsilon = \Pi_{i=0}^{n} f_i/(f_i + w_i)$, and corresponding probability of success is $1 - \epsilon$. We can now compute payoffs for both defender and attacker:

$$(1-\epsilon) * V_\Theta^d(a_i) + \epsilon * V_\Theta^a(a_i) + \sum_{Q_i^\Psi \in \sigma_\Psi, o_j \in \gamma_i} C(a_i, o_j) * Q_i^\Psi \quad (3)$$

$$(1-\epsilon) * V_\Psi^d(a_i) + \epsilon * V_\Psi^a(a_i) - \sum_{Q_i^\Psi \in \sigma_\Psi, o_j \in \gamma_i} C(a_i, o_j) * Q_i^\Psi \quad (4)$$

We note that it is also possible to quickly detect situations where the attacker cannot possibly succeed because the number of operations circumvented for some type is less than the number of operations

run by the defender for this type. In these cases, the above equations simplify to:

$$V^d_\Theta(a_i) + \sum_{Q^\Psi_i \in \sigma_\Psi, o_j \in \gamma_i} C(a_i, o_j) * Q^\Psi_i \quad (5)$$

$$V^d_\Psi(a_i) - \sum_{Q^\Psi_i \in \sigma_\Psi, o_j \in \gamma_i} C(a_i, o_j) * Q^\Psi_i \quad (6)$$

Looking back at Table 1 we can provide some additional insight into why this compact representation works. Notice that regardless of whether the defender chooses operation o_1 or o_2 he will receive identical payoffs. For example, if the defender chooses operation o_1 the reward value if the attacker just avoids o_1 is -8 and if the attacker just avoids o_2 it is 4. Similarly, if the defender chooses operation o_2 their reward value if the attacker just avoids o_2 is -8 and if the attacker just avoids o_1 it is 4. Given that the attacker's strategy is to optimize against the defender's strategy and that resources are split equally among o_1 and o_2, the attacker is indifferent between avoiding just o_1 or just o_2 since both yield identical payoffs.

6. EXTENSION TO MULTIPLE LEVELS OF PROTECTION

Up to this point we have assumed that each operation affects exactly one area, and that every operation is able to prevent any attack if it is not circumvented by the attacker. In this section we relax these assumptions and allow for a more general model of the effects of operations on the success or failure of an attack. We allow each operation to affect an arbitrary number of areas, and to prevent attacks in each area with a different probability. The ability to represent operations that affect different areas is useful for representing patrols in adjacent areas, or for representing security measures that may not be directly applicable to a single physical area, but has a broad effect across many different areas.

We define a function $S(a_i, o_j) \in [0 \ldots 1]$ that expresses the probability that operation o_j will prevent an attack in area a_i. A value of 0 represents an operation that has no effect on a particular area, and a value of 1 represents perfect protection. As before, any operation can be circumvented by the adversary to mitigate the protective effect of the operation. The main difference in this model is that we must now consider a definition of operation types that accounts for the effectiveness of operations in different areas. Operations may only be collapsed in the compact representation if they provide identical coverage in every area. Given that restriction, we can extend Proposition 1 by a similar argument (omitted here) to show that it is optimal to randomize uniformly across operations that are identical in this respect.

Extending our model to this more comprehensive model in both the full representation and compact representation requires only a minimal change to the payoff calculations. Specifically, the total probability of successfully preventing an attack is computed by multiplying together the $1 - S(a_i, o_j)$ values for all of the operations in σ_Θ that are not circumvented in σ_Ψ for the area a_i that is attacked and then subtracting this value from 1. Specifically, the multiplication of these values, $1 - S(a_i, o_j)$, represents the chance that all operations failed to catch the adversary and the chance of success is easily determined by subtracting this value from 1. Denoting this value by Z we have the revised equations:

$$(1 - Z) * V^d_\Theta(a_i) + Z * V^a_\Theta(a_i) + \sum_{o_i \in \sigma_\Psi} C(a_i, o_i) \quad (7)$$

$$(1 - Z) * V^d_\Psi(a_i) + Z * V^a_\Psi(a_i) - \sum_{o_i \in \sigma_\Psi} C(a_i, o_i) \quad (8)$$

Computing the payoffs for the compact representation in this case requires one additional manipulation. First, we must compute the probability that each operation is circumvented for each set of identical operations γ_i, based on the attacker strategy. This probability of circumvention is factored into the computation of the overall probability of capture by scaling each $S(a_i, o_j)$ in the computation by the probability that o_j will be circumvented by the attacker. Given this scaling term, the process for computing the payoffs is the same as described previously.

7. EVALUATION

In this section we provide empirical results to demonstrate the benefits of our compact representation on scalability. The effectiveness of this representation depends primarily on the number of unique types of operations that are present in the original game; in the worst case every operation is unique in some way and in that case the compact representation is identical to the full representation. Our compact representation is most effective in cases where each operation affects relatively few areas, and the effectiveness of operations (in terms of the probability of preventing an attack) can be categorized into a small number of discrete levels of protection. This maximizes the chance that there will be identical operations which can be merged in the compact representation. In principle it would also be possible to merge similar operations with some loss of solution quality, but we defer investigation of this method for approximation to future work.

We present simulation results focusing on the computational efficiency of our methods, and particularly the benefits of the compact representation in cases where there are identical operations. All experiments are run on a system with an Intel 2 GHz processor and 1 GB of RAM. We used a publicly available linear programming package called GLPK to solve optimization problems as specified in the original DOBSS procedure. The solver was allowed to use up to 700 MB of memory during the solution process. For larger game instances, solving the problem with the full representation runs out of memory and solutions cannot be found. In the results presented below we exclude results for cases where the full representation was not able to produce a result using the allotted memory.

To test the solution methods we generate random game instances by randomly selecting payoff values and the circumvention costs for each area. For each experiment we generated 20 random game instances and averaged the results (there is little variance in the runtimes for different problem instances). We consider three different scenarios. The first scenario shown in Figure 2 has a single area, and the defender is allowed to allocate up to 5 resources to run operations. We increase the number of different operations available to protect this area along the x-axis. For the compact representation we vary the number of unique types of operations to show how this impacts the efficiency of the solution method. Results are shown for 1, 2, and 4 unique operation types, however, we only show the results in the 4 unique operation types case up to 10 operations. It is clear that more operations in this case would have taken a substantial amount of time. As shown in Figure 2, the full representation is

unable to find a solution within the memory limit for games with 8 or more operations, while the compact representation is able to run up to 20 operations in less than 1 second in the ideal case where there is a single operation type.

The next scenario presents results for the case where there is an increasing number of areas, and each area has exactly 3 operations associated with it. There are 5 resources available for the defender, and each operation provides maximum protection for the area it is associated with. This implies that there is one unique operation type for every area. Examining Figure 3, we show the improvement of our compact representation over the full representation. For more than 4 areas, the full representation failed to achieve a solution within the memory bounds. For 5 areas, the compact representation runs much faster than the full representation, with a total runtime of less than 1 second versus the 177 seconds required by the full representation to find a solution for the case with only 4 areas. Even if the number of operations associated with each area is a relatively small constant our compact representation provides substantial benefits. As the number of similar operations associated with an area increases, this advantage grows (as shown in our first experiment).

Finally we consider a scenario where operations are distributed randomly across possible areas. Again, each operation is associated with a single area. The total number of operations is set similarly to the previous experiment, in that that the total number of operations is three times the number of areas. However, we randomly assign operations to areas (with each area having at least one operation) so the number is no longer uniform. Once again the defender has 5 resources available and each operation provides full protection to the area it is associated with. Looking at Figure 4, we see similar benefits for the compact representation in this case as in the previous experiment with a uniform distribution of operations.

These results show the potential benefits of the compact representation in improving scalability by exploiting similarities in the effects of some operations. We have shown that the most important factor is the number of unique types of operations that exist and how many of these operations there are. If the number of types is low the compact representation performs efficiently.

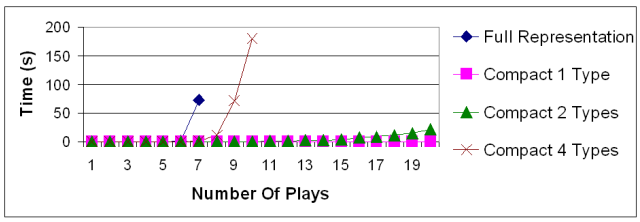

Figure 2: Runtime: Increasing number of operations with 1 target and 5 resources

8. CONCLUSION

Allocating resources to defend critical infrastructure, high profile events, and transportation systems among other things remains an important problem in many security domains. While there are a number of methods in use today for addressing this problem, one notable approach that is increasingly finding more use in security applications is that of game theory. In fact, game theory has seen successful application at the Los Angeles International Airport and for the United States Federal Air Marshals Services [8, 14].

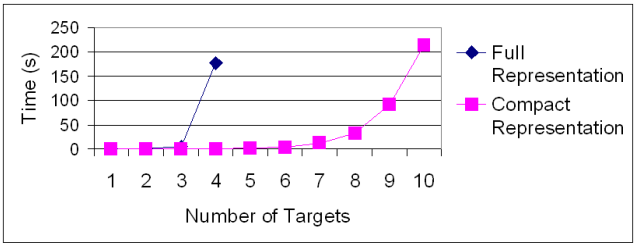

Figure 3: Runtime: Increasing areas with 5 resources and 3 operations per area

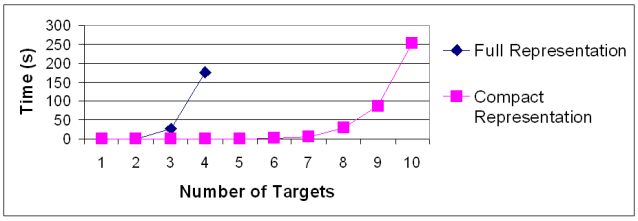

Figure 4: Runtime: Increasing areas with operations randomly distributed and 5 resources

We introduce a new form of security game that extends previous models in several important directions. First, this model includes a more fine-grained representation for the defender's strategy space, explicitly considering both the location of a security activity and the type of activity that a security resource will perform. Second, we allow for different levels of effectiveness for different security actions, including the possibility that an activity has different effects across multiple locations. Previous models have also assumed that all of the possible attack strategies for the attacker are known with certainty, which is unrealistic in real-world security problems. In particular, attackers can often adapt to circumvent specific known security measures. We extend the security game model on the attacker's side to handle this possible in a generic way, allowing attackers to circumvent specific security activities at some cost. This leads to an increased value for randomness and unpredictability in the defender's strategy, even among actions that may be similar in terms of the areas they affect. Solving this new class of games presents a computational challenge that existing solution methods are not able to handle. We address this by introducing a compact representation for these games that exploits symmetries between similar types of security activities, and provide experimental results showing the resulting improvements in runtime.

9. REFERENCES

[1] N. Agmon, V. Sadov, S. Kraus, and G. Kaminka. The impact of adversarial knowledge on adversarial planning in perimeter patrol. In *AAMAS*, 2008.

[2] L. Babu, L. Lin, and R. Batta. Passenger grouping under constant threat probability in an airport security system. *European Journal of Operational Research*, 168:633–644, 2006.

[3] N. Basilico, N. Gatti, and F. Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topology. In *AAMAS*, 2009.

[4] G. Brown, M. Carlyle, J. Kline, and K. Wood. A two-sided optimization for theater ballistic missile defense. *Operations Research*, 53:263–275, 2005.

[5] G. Brown, M. Carlyle, J. Royset, and K. Wood. On the

complexity of delaying an adversary's project. *The Next Wave in Computing, Optimization and Decision Technologies*, pages 3–17, 2005.

[6] G. Brown, M. Carlyle, J. Salmerón, and K. Wood. Defending critical infrastructure. *Interfaces*, 36(6):530–544, 2006.

[7] N. Gatti. Game theoretical insights in strategic patrolling: Model and algorithm in normal-form. In *ECAI*, 2008.

[8] C. Kiekintveld, M. Jain, J. Tsai, J. Pita, M. Tambe, and F. Ordóñez. Computing Optimal Randomized Resource Allocations for Massive Security Games. In *AAMAS*, 2009.

[9] R. C. Larson. A hypercube queueing model for facility location and redistricting in urban emergency services. *Computers and OR*, 1(1):67–95, 1974.

[10] K. Lye and J. Wing. Game strategies in network security. *International Journal of Information Security*, 4(1-2):71–86, 2005.

[11] X. Nie, R. Batta, C. Drury, and L. Lin. Optimal placement of suicide bomber detectors. *Military Operations Research*, 12:65–78, 2007.

[12] P. Paruchuri, J. Marecki, J. Pearce, M. Tambe, F. Ordóñez, and S. Kraus. Playing games for security: An efficient exact algorithm for solving bayesian stackelberg games. In *AAMAS*, 2008.

[13] P. Paruchuri, M. Tambe, F. Ordonez, and S. Kraus. Security in multiagent systems by policy randomization. In *AAMAS*, 2006.

[14] J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, and S. Kraus. Deployed armor protection: The application of a game theoretic model for security at the los angeles international airport. In *AAMAS*, 2008.

[15] S. Ruan, C. Meirina, F. Yu, K. R. Pattipati, and R. L. Popp. Patrolling in a stochastic environment. In *10th Intl. Command and Control Research and Tech. Symp.*, 2005.

[16] T. Sandler and D. Arce. Terrorism and game theory. *Simulation and Gaming*, 34(3):319–337, 2003.

[17] V. Srivastava, J. Neel, A. MacKenzie, R. Menon, L. Dasilva, J. Hicks, J. Reed, and R. Gilles. Using game theory to analyze wireless ad hoc networks. *IEEE Communications Surveys and Tutuorials*, 7(4), 2005.

[18] L. Wein. Homeland security: From mathematical models to policy implementation. In *Operations Research*, 2008.