International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

The Evolution from Physical Protection to Cyber Defense

Pavan Navandar

Cybersecurity SAP Security Engineer, Tata Consultancy Services, USA

ABSTRACT: Security is a critical concern around the world. In many areas, from cybersecurity to sustainability, limited security resources always prevent complete security coverage. Instead, these limited resources must be scheduled (or distributed or deployed), while simultaneously considering the importance of different targets, the responses of the adversaries to the security posture, and the potential uncertainties in adversary payoffs and observations, etc. Computational game theory can help generate such security schedules. Indeed, casting the problem as a Stackelberg game, we have developed new algorithms that are now deployed over multiple years in multiple applications for scheduling of security resources. These applications are leading to real-world use-inspired research in the emerging research area of "security games." The research challenges posed by these applications include scaling up security games to real-world-sized problems, handling multiple types of uncertainty, and dealing with bounded rationality of human adversaries. In cybersecurity domain, the interaction between the defender and adversary is quite complicated with high degree of incomplete information and uncertainty. While solutions have been proposed for parts of the problem space in cybersecurity, the need of the hour is a comprehensive understanding of the whole space including the interaction with the adversary. We highlight the innovations in security games that could be used to tackle the game problem in cybersecurity.

KEYWORDS: Cyber Defense Evolution, Physical Security, Security Games, Stackelberg Game Theory, Adversarial Modeling, Uncertainty in Cybersecurity, Resource Optimization

I. INTRODUCTION

Security is a critical concern around the world that manifests in problems such as protecting our cyber infrastructure from attacks by criminals and other nation-states; protecting our ports, airports, public transportation, and other critical national infrastructure from terrorists; protecting our wildlife and forests from poachers and smugglers; and curtailing the illegal flow of weapons, drugs, and money across international borders. In all of these problems, there are limited security resources which prevents security coverage of all the targets at all times; instead, security resources must be deployed intelligently taking into account differences in the importance of targets, 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 models interactions among multiple self-interested agents, is well-suited to the adversarial reasoning required for the security resource allocation and scheduling problem. Casting the physical problem as a Stackelberg game, we have developed new algorithms for efficiently solving such games that provide randomized patrolling or inspection strategies. These algorithms have led to successes and advances over previous human-designed approaches in security scheduling and allocation by addressing the key weakness of predictability in human-designed schedules. These algorithms are now deployed in multiple applications. The first application was A rumor (Assistant for Randomized Monitoring over Routes), which was deployed at the Los Angeles International Airport (LAX) in 2007 to randomize checkpoints on the roadways entering the airport and canine patrol routes within the airport terminals [1]. Following that, came several other applications: I rise (Intelligent Randomization In Scheduling), a game-theoretic scheduler for randomized deployment of the US Federal Air Marshals, has been in use since 2009 [1]; P protect , which schedules the US Coast Guard's (USCG) randomized patrolling of ports, has been deployed in the port of Boston since April 2011 and is in use at the port of New York since February 2012 [2] and has spread to other ports such as Los Angeles/Long Beach, Houston, and others; another application for deploying escort boats to protect ferries has been deployed by the USCG since April 2013 [3]; and T rusts (Tactical Randomization for Urban Security in Transit Systems) [4], which has been evaluated in field trials by the Los Angeles Sheriff's Department (LASD) in the LA Metro system. Most recently, P was —another game-theoretic application was tested by rangers in Uganda for protecting wildlife in Queen Elizabeth National Park (QENP) in April 2014 [5]; M idas was tested by the USCG for protecting fisheries [6]. These initial successes point the way to major future applications in a wide range of security domains. Indeed, researchers have started to explore the use of security game models in tackling security issues in the

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

cyber world, such as deep packet inspection [7], optimal use of honey pots [8], and enforcement of privacy policies [9 , 10].

Given the many game-theoretic applications for solving real-world security problems, this article provides an overview of the models and algorithms, key research challenges, and a brief description of our successful deployments. We also provide an overview of applying Stackelberg game-based models to cybersecurity and compare with other existing approaches to model defender–adversary interaction in cybersecurity. Overall, the work in security games has produced numerous game-theoretic decision aids that are in daily use by security agencies to optimize their limited security resources. The implementation of these applications required addressing fundamental research challenges and has led to an emerging "science of security games" consisting of a general framework for modeling and solving security resource allocation problems. We categorize the research challenges associated with security games into four broad categories: (I) addressing scalability across a number of dimensions of the game, (ii) tackling different forms of uncertainty that be present in the game, (iii) addressing human adversaries' bounded rationality, and (iv) evaluation of the framework in the field. Given the success in providing solutions for many security domains involving the protection of critical infrastructure, the science of security games has evolved and expanded to include new types of security domains for wildlife and environmental protection. These "green security games" address important global conservation problems and introduce additional research challenges that require incorporating new techniques such as planning and learning into security games. The issues in cybersecurity provide an even richer set of challenges that include partial observability and deception.

The rest of the article is organized as follows: "Stackelberg Security Games" section introduces the general security games model, "Addressing scalability in real-world problems" section describes the approaches used to tackle scalability issues, "Addressing uncertainty in real-world problems" section describes the approaches to deal with uncertainty, "Addressing bounded rationality in real-world problems" section focuses on bounded rationality, "Addressing field evaluation in real-world problems" section provides details of field evaluation of the science of security games and "Cybersecurity: challenges and opportunities" section describes some approaches of applying security game models to cybersecurity and privacy and also other game-theoretic approaches to cybersecurity.

II. RELATED WORK

Stackelberg security games

Stackelberg games were first introduced to model leadership and commitment [11]. The term Stackelberg security games (SSG) were first introduced by Kiekintveld *et al.* [12] to describe specializations of a particular type of Stackelberg game for security as discussed below. This section provides details on the 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.

SSG Model

In SSG, a defender must perpetually defend a set of targets T using a limited number of resources, whereas the attacker is able to surveil and learn the defender's strategy and attack after careful planning. An action, or "pure strategy," for the defender represents deploying a set of resources R 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 "mixed strategy" of the defender is a probability distribution over the pure strategies. 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.

A key assumption of SSG (we will sometimes refer to them as simply security games) is that the payoff of an outcome depends only on the target attacked, and whether or not it is "covered" (protected) by the defender [12]. The payoffs do "not" depend on the remaining aspects of the defender allocation. 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 unsuccessful attack and are sufficient to define the utilities for both players for all possible outcomes in the security domain. More formally, if target t is attacked, the defender's utility is if t is covered, or if t is not covered. The attacker's utility is if t is covered, or if t is not covered. Table 1 shows an example of security game with two targets, t_1 and t_2. In this example game, if the defender was covering target t_1 and the attacker attacked t_1, the defender

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

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, while 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 general security games are not necessarily zero-sum.

Table 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

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. The defender's mixed strategy is a probability distribution over all pure strategies, where a pure strategy is an assignment of the defender's limited security resources to targets. This solution concept is known as a Stackelberg equilibrium [13].

The most commonly adopted version of this concept in related literature is called strong Stackelberg equilibrium (SSE) [14–17]. In security games, the mixed strategy of the defender is equivalent to the probabilities that each target *t* is covered by the defender, denoted by [18]. Furthermore, it is enough to consider a pure strategy of the rational adversary [15], which is to attack a target *t*. The expected utility for defenders for a strategy profile (C, *t*) is defined as, and a similar form for the adversary. An SSE for the basic security games (non-Bayesian, rational adversary) is defined as follows.

Definition 1.
A pair of strategies form an SSE if they satisfy the following:
1. *The defender plays a best response: for all defender's strategy C where t(C) is the attacker's response against the defender strategy C.*
2. *The attacker plays the best response: for all target t.*
3. *The attacker breaks ties in favor of the defender: for all targets such that.*

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 [17]. 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 [19–21] and have used these robust solution concepts in some of the later applications.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

In the following sections, we present three key challenges in solving real-world security problems which are summarized in Fig. 1 : (I) scaling up to real-world-sized security problems, (ii) handling multiple uncertainties in security games, and (iii) dealing with bounded rationality of human adversaries. While Fig. 1 does not provide an exhaustive overview of all research in SSG, it provides a general overview of the areas of research. In each case, we will use a domain example to motivate the specific challenge and then outline the key algorithmic innovation needed to address the challenge.

	Scalability				Uncertainty		Attacker Bounded Rationality
Challenge	Large defender strategy space	Large defender & attacker strategy spaces	Mobile resources & moving targets	Multiple boundedly rational attackers	Unifications of uncertainties	Dynamic execution uncertainty	Learning attacker behaviors
Domain Example	Federal Air Marshals Service	Road Network Security	Ferry Protection	Fishery Protection	Security in LAX Airport	Security in Transit System	Green Security Domains: wildlife/ fishery protection
Algorithmic Solution	ASPEN: strategy generation approach	RUGGED: double oracle approach	CASS: compact representation of strategy	MIDAS: cutting plane approach	URAC: multi-dimensional reduction & divide-and-conquer approach	Markov Decision Processes approach	Behavioral models & Human subject experiments

Figure 1.
Summary of real-world security challenges.

Addressing scalability in real-world problems

For simple examples of security games, such as the one shown in the earlier section, the SSE can be calculated by hand. However, as the size of the game increases, hand calculation is no longer feasible and an algorithmic approach for generating the SSE becomes necessary. Conatser and Sandholm [15] provided the first complexity results and algorithms for computing optimal commitment strategies in Bayesian Stackelberg games, including both pure and mixed-strategy commitments. An improved algorithm for solving Bayesian Stackelberg games, Decomposed Optimal Bayesian Stackelberg Solver (D boss) [16], is central to the fielded application A rmor in use at the LAX [1]. These early works required that the full set of pure strategies for both players be considered when modeling and solving SSG. However, many real-world problems feature billions of pure strategies for either the defender and/or the attacker. Such large problem instances cannot even be represented in modern computers, let alone solved using previous techniques.

In addition to large strategy spaces, there are other scalability challenges presented by different real-world security domains. There are domains where, rather than being static, the targets are moving and thus the security resources need to be mobile and move in a continuous space to provide protection. There are also domains where the attacker may not conduct careful surveillance and planning that is assumed for an SSE and thus it is important to model the bounded rationality of the attacker in order to predict their behavior. In the former case, both the defender and attacker's strategy spaces are infinite. In the latter case, computing the optimal strategy for the defender given attacker behavioral (bounded rationality) model is computationally expensive. In this section, we thus highlight the critical scalability challenges faced to bring SSG to the real world and the research contributions that served to address these challenges.

Scale-up with large defender strategy spaces.

This section provides an example of a research challenge in security games where the number of defender strategies is too enormous to be enumerated in computer memory. In this section as in others that will follow, we will first provide a domain example motivating the challenge and then the algorithmic solution for the challenge.

Domain example rise for US Federal Air Marshals Service

The US Federal Air Marshals Service (FAMS) distributes air marshals to flights departing from and arriving in 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 cities, and international flights from different countries. Security resource allocation in this domain is significantly more challenging than for A rumor: a limited number of air marshals need to be scheduled to cover thousands of commercial flights each day.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

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. For an example scenario with 1000 flights and 20 marshals, there are over 10^{41} possible schedules that could be considered. Yet 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 [22]. 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 New York to London and back.

Against this background, the I rise system has been developed and deployed by FAMS since 2009 to randomize schedules of air marshals on international flights. In I rise, the targets are the set of 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, D boss is no longer applicable to the FAMS domain. Instead, I use the much faster A spen algorithm [23] to generate the schedule for thousands of commercial flights per day.

Algorithmic solution—incremental strategy generation (A spen)

In this section, we describe one particular algorithm A spen , that computes SSEs in domains with a "very large" number of pure strategies (up to billions of actions) for the defender [23]. These pure strategies can be represented as integral points in a high-dimensional space. A spen builds on the insight that 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 (applying Arthrodire theorem [24] to the convex hull of pure strategies). A spen exploits this by using a "column generation"- based approach [25] for the defender, in which defender pure strategies are iteratively generated and added to the optimization formulation.

In A spen 's security game, the attacker can choose any of the flights to attack, and each air marshal can cover one schedule. Each schedule here is a possible 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. For example, would be a flight schedule, where t_1 is an outbound flight and t_2 is an inbound flight for one air marshal. 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. Thus, e.g., if there are two air marshals, one possible joint schedule would be, where the first air marshal covers flights t_1 and t_2, and the second covers flights t_3 and t_4. As mentioned previously, A spent employs incremental strategy (column) generation since all the defender pure strategies cannot be enumerated for such a massive problem. A spent decomposes the problem into a "master" problem and a "slave" problem, which are then solved iteratively. Given a number of pure strategies, the master solves the optimization problem for the defender and the attacker with these pure strategies, whereas the slave is used to generate a new pure strategy for the defender in every iteration. "This incremental, iterative strategy generation process allows A spent to avoid generation of the entire set of pure strategies." In other words, by exploiting the small support size mentioned above, only a few pure strategies get generated via the iterative process; and yet we are guaranteed to reach the optimal solution.

The iterative process is graphically depicted in Fig. 2 . The master operates on the pure strategies (joint schedules) generated thus far, which are represented using the matrix. Each column of, is one pure strategy (or joint schedule). An entry Pi_n in the matrix is 1 if a target t_{it} is covered by joint-schedule, and 0 otherwise. For example, in Fig. 2 , the joint schedule J_3 covers target t_1 but not target t_2. The objective of the master problem is to compute the optimal mixed strategy of the defender over the pure strategies in. The objective of the slave problem is to generate the best joint schedule (pure strategy) to add to. 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 [23]). While a naïve approach would be to iterate over all possible pure strategies to find the pure strategy with the maximum potential, A spen formulates the slave problem as a minimum-cost integer flow problem to efficiently identify the best pure strategy to add. Spend always converges on the optimal mixed strategy for the defender.

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

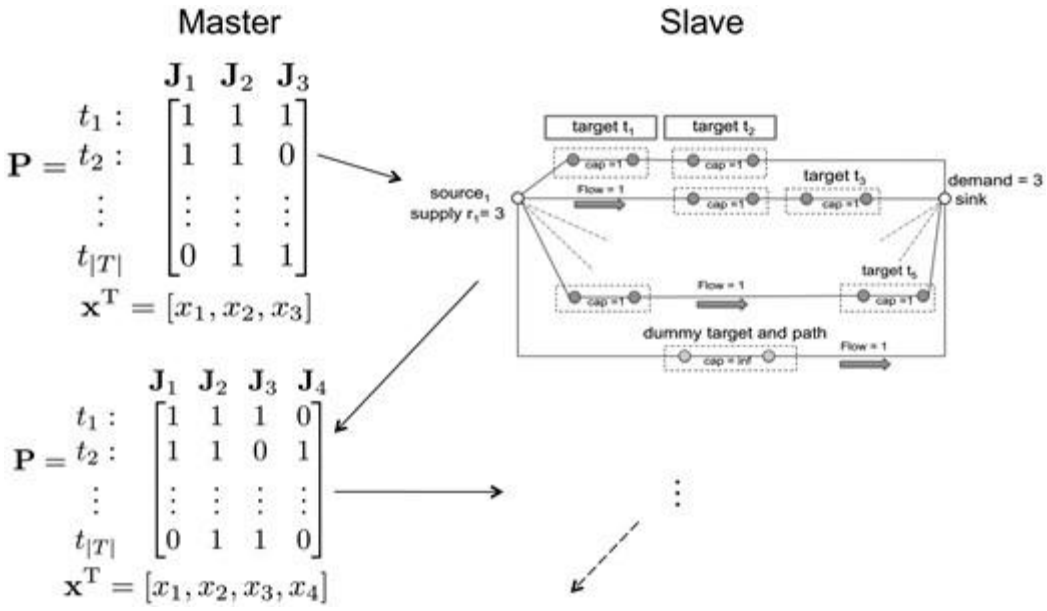


Figure 2.
Strategy generation employed in A spen: the schedules for a defender are generated iteratively. The "slave" problem is a novel minimum-cost integer flow formulation that computes the new pure strategy to be added to; is computed and added in this example.

Employing incremental strategy (column) generation for large optimization problems is not an "out-of-the-box" approach; the problem must be formulated in a way that allows for domain properties to be exploited. The novel contribution of A spen 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.

Scale-up with large defender and attacker strategy spaces.
Whereas the previous section focused on domains where only the defender's strategy was difficult to enumerate, we now turn to domains where both defender and attacker strategies are difficult to count. Once again, we give a domain example and then an algorithmic solution.

Domain example—road network security
One area of immense 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 [26], the Mumbai police 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, e.g., 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—e.g., 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. For example, in a fully connected graph with 20 nodes and 190 edges, the number of defender pure strategies for only 5 defender resources is or almost 2 billion, while the number of attacker pure strategies (i.e., paths without cycles) is on the order of 10^{18}. Real-world networks are significantly larger, e.g., the entire road network of the city of Mumbai has 9503 nodes (intersections) and 20 416 edges (streets), and the security forces can deploy dozens (but not as many as number of edges) of resources. 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 [27].

Algorithmic solution—double oracle incremental strategy generation (R urged)
In domains such as the urban network security setting, the number of pure strategies of both the defender and the attacker are exponentially large. In this section, we describe the urged algorithm [28], which generates pure strategies

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

for both the defender and the attacker. This algorithm is inspired by the double oracle algorithm of solving large-scale games [29].

R urged 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 3 shows the working of R urged : at each iteration, the Minimax module generates the optimal mixed strategies 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 , and the Best Response Attacker module generates a new strategy for the attacker that is a best response against the defender's current strategy . The rows X 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, columns A_j are the pure strategies for the attacker; they are the attack paths in the urban road network domain. The values in the matrix represent the payoffs to the defender. For example, in Fig. 3 , the row denoted by X_1 indicates that there was one checkpoint setup, and it provides a defender payoff of −5 against attacker strategy (path) A_1 , and a payoff of 10 against attacker strategy (path) A_2 .

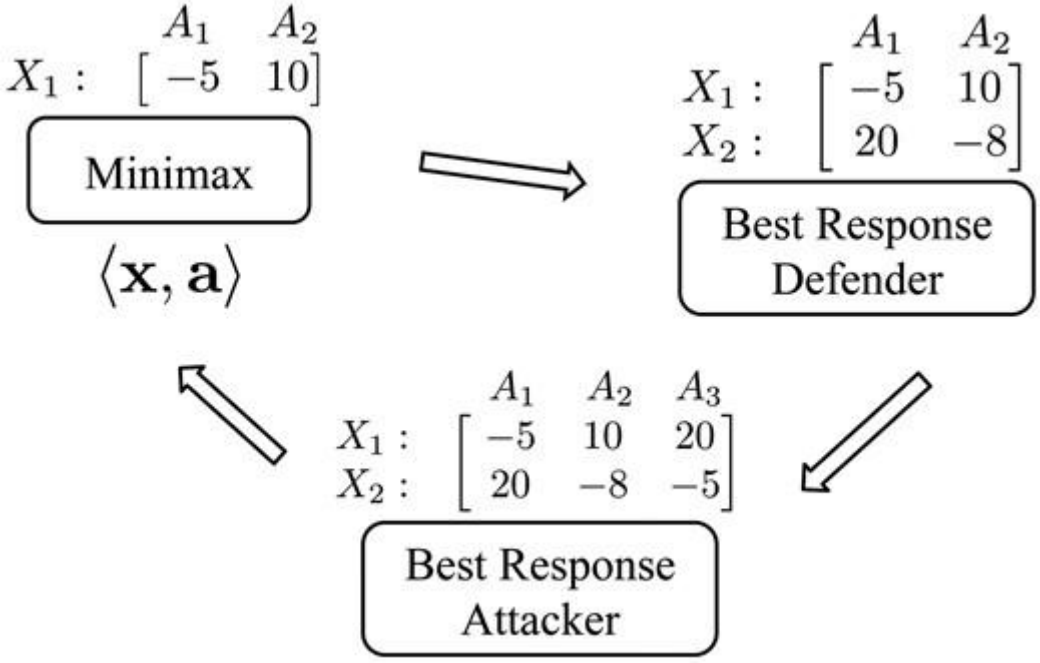


Figure 3.
Strategy generation employed in R urged: the pure strategies for both the defender and the attacker are generated iteratively.

In Fig. 3 , we show that R urged iterates over two oracles: the defender best response and the attacker best response. In this case, the defender's best response oracle has added a strategy X_2, and the attacker's best response oracle then adds a strategy A_3. The algorithm stops when neither of the generated best responses improves on the current minimax strategies.

The contribution of R urged is to provide the mixed integer formulations for the best response modules which enable the application of such a strategy generation approach. The key once again is that R urged is able to converge to the best solution without enumerating the entire space of defender and attacker strategies. However, originally R urged could only compute the optimal solution for deploying up to four resources in real-city network with 250 nodes within a 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 [28]). More recent work [27] builds on R urged and proposes S nares , which allows scale-up to the entire city of Mumbai, with 10–15 checkpoints.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

Scale-up with mobile resources and moving targets.
Whereas the previous two sections focused on incremental strategy generation as an approach for scale-up, this section introduces another approach: use of compact marginal probability representations. This alternative approach is shown in use in the context of a new application for protecting ferries.

Domain example—ferry protection for the USCG
The USCG handles protecting domestic ferries, including the Staten Island Ferry in New York, from potential terrorist attacks. Here 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 (Figure 4). However, the numbers of patrol boats are often limited, thus the defender cannot always protect the ferries and locations. We thus 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 [3].

Figure 4.
Escort boats protecting the Staten Island Ferry use strategies generated by our system.

The key research challenge is the fact that the ferries are continuously 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 (in the sense that the solution quality provided by our work provides a lower bound), discretizing the attacker's strategy space would result in loss of utility (in the sense that this would provide only an upper bound, and thus an unreliable guarantee of true solution quality). 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.

Algorithmic solution—compact strategy representation (C ass)
In this section, we describe the C ass (Solver for Continuous Attacker Strategy) algorithm [3] 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

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

the defender and attacker are continuous with respect to the continuous spatial and time components of the problem domain. The C ass 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 that exploits the piecewise-linear structure of the attacker's expected utility function. The insight of C ass is to compactly represent the defender's mixed strategies as a "marginal" probability distribution, overcoming the shortcoming of an exponential number of pure strategies for the defender.

C ass casts problems such as the ferry protection problem mentioned above 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 locations of the targets vary with respect to 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 C ass, the defender's strategy space is discretized, and her mixed strategy is compactly represented using flow distributions. Figure 5 shows an example of a ferry transition graph in which each node of the graph indicates a particular pair (location, 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 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.

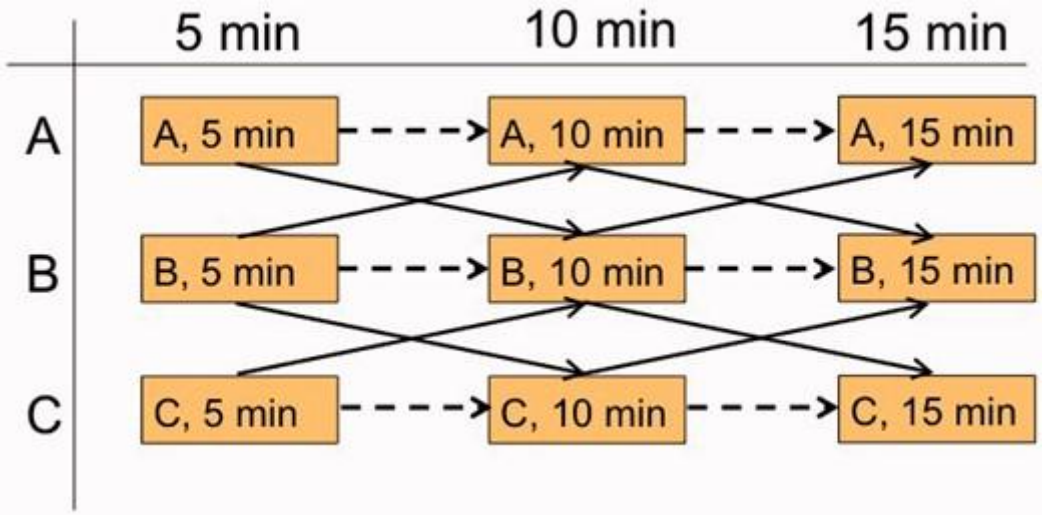


Figure 5.
An example of a ferry transition graph.

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., where N is the number of location points and T is the number of time steps. To address this challenge, C ass proposes a compact representation of the defender's mixed strategy. Instead of directly computing a probability distribution over pure strategies for the defender, C ass 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). C ass 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 C ass to reformulate

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

the resource-allocation problem as computing the optimal "marginal" coverage of the defender over a number of *O* (NT) edges of the transition graph.

Scale-up with boundedly rational attackers.
One key challenge of real-world security problems is that the attacker is boundedly rational; the attacker's target choice is nonoptimal. In SSGs, attacker-bounded rationality is often modeled via behavior models such as Quantal Response (QR) [30 , 31]. In general, QR attempts to predict the probability the attacker will choose each target with the intuition is that the higher the expected utility at a target is, the more likely that the adversary will attack that target. Another behavioral model that was recently shown to provide higher prediction accuracy in predicting the attacker's behavior than QR is Subjective Utility QR (SUQR) [32]. SUQR is motivated by the lens model which suggested that evaluation of adversaries over targets is based on a linear combination of multiple observable features [33]. However, handling multiple attackers with these behavioral models in the context of large defender's strategy space is computational challenge. In this section, we mainly focus on handling the scalability problem given behavioral models of the attacker. The problem of handling the attacker's bounded rationality (e.g., modeling and learning) is explained in detail in "Addressing bounded rationality in real-world problems" section.

To handle the problem of large defender's strategy space given behavioral models of attackers, we introduce yet another technique of scaling up, which is similar to the incremental strategy generation. Instead, here we use incremental marginal space refinement. We use the compact marginal representation, discussed earlier, but refine that space incrementally if the solution produces violates the necessary constraints.

Domain example—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. The 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.

Natural resource conservation domains such as fishery protection raise a number of new research challenges. 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 over time. Third, the adversaries are less strategic given the short planning windows between actions.

III. EXPERIMENTAL RESULTS

Algorithmic solution—incremental constraint generation (Midas)
Generating effective strategies for domains such as fishery protection requires an algorithmic approach which is both "scalable" and "robust". For scalability, the defender handles 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 must protect against "multiple" boundedly rational adversaries. Bounded rationality models, such as the 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 can 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 nonlinear models resulting in a computationally difficult optimization problem for the defender. Combining these factors, M Idas models a population of boundedly rational adversaries and utilizes available data to learn the behavior models of the adversaries using the SUQR model in order to improve the way the defender allocates its patrolling resources.

Previous work on boundedly rational adversaries has considered the challenges of scalability and robustness separately, in [34 , 35] and [5 , 6], respectively. The M Idas algorithm was introduced to merge these two research threads for the first time by addressing scalability and robustness simultaneously. Figure 6 provides a visual overview of how M Idas operates as an iterative process. Like the A spen algorithm described earlier, given the sheer complexity of the

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

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. Second, the complex spatiotemporal constraints associated with patrols are initially ignored and then incrementally added back using cut generation. In other words, we ignore the spatiotemporal constraint that a patroller cannot simply appear and disappear at different locations instantaneously; and that a patroller must pass through regions connecting two different regions if the patroller is going from one region to another. This significantly simplifies the master problem.

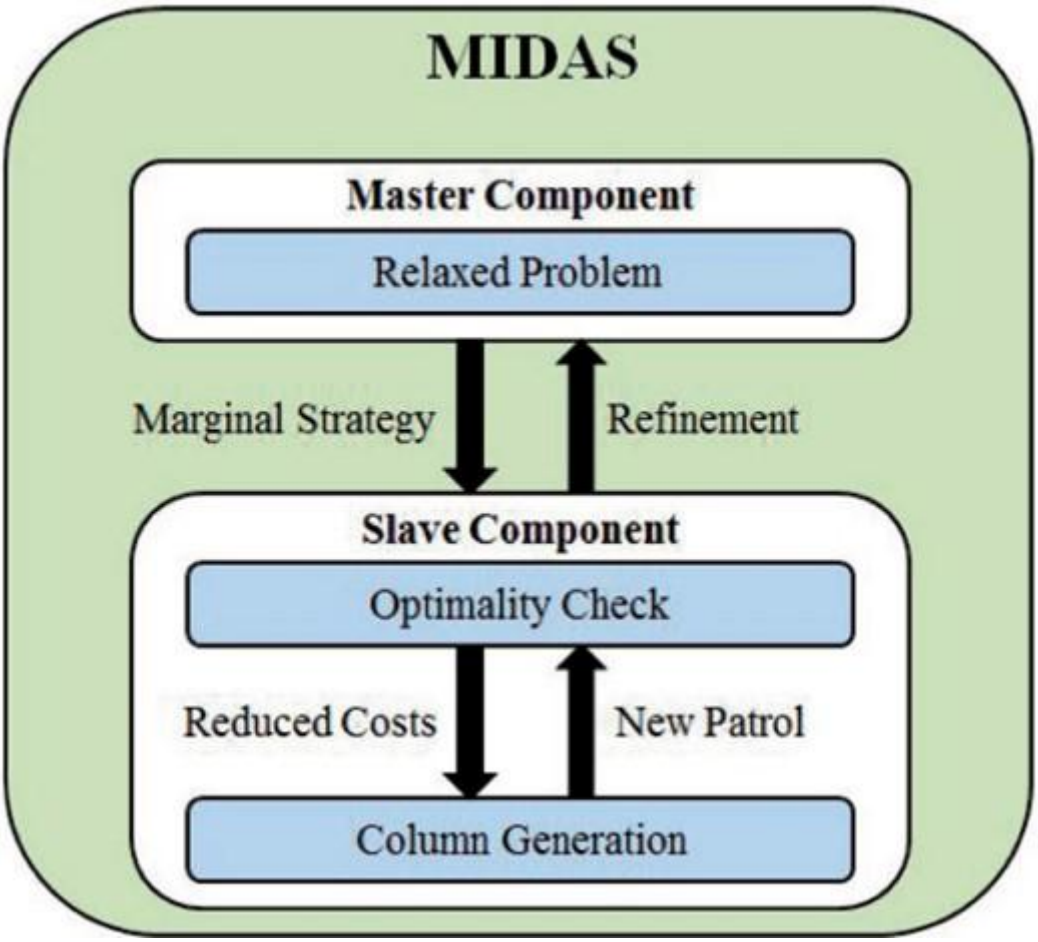


Figure 6.
Overview of the multiple iterative process within the M Idas algorithm.

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 be expressed as a linear combination, i.e., probability distribution, of patrols. 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., [23], 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. Finally, M Idas has been successfully deployed and evaluated by the USCG in the Gulf of Mexico.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

Addressing uncertainty in real-world problems

The standard security game model features a number of strong assumptions including that the defender has perfect information about the game payoff matrix as well as the attacker's behavioral model. Additionally, the defender is assumed to be capable of executing the computer patrolling strategy. However, uncertainty is endemic in real-world security domains and thus it may be impossible or impractical for the defender to accurately estimate various aspects of the game. Also, there are many numbers of practicalities and unforeseen events that may force the defender to change their patrolling strategy. These types of uncertainty can significantly deteriorate the effectiveness of the defender's strategy and thus addressing uncertainty when generating strategies is a key challenge of solving real-world security problems. This section describes several approaches for dealing with various types of uncertainties in SSGs.

We first summarize the major types of uncertainties in SSGs as a 3-dimensional uncertainty space with the following three dimensions (Fig. 7): (I) uncertainty in the adversary's payoffs; (ii) uncertainty related to the defender's strategy (including uncertainty in the defender's execution and the attacker's observation); and (iii) 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 7 also shows existing algorithms for addressing uncertainty in SSGs which follow the two different approaches. First approach is applying robust optimization techniques using uncertainty intervals to represent uncertainty in SSGs. For example, Brass [36] is a robust algorithm that only addresses attacker-payoff uncertainty, R econ [19] is another robust algorithm that focuses on addressing defender-strategy uncertainty, and Monotonic Maximin [37] is to handle the uncertainty in the attacker's bounded rationality. Finally, U race (Unified Robust Algorithmic framework for addressing Uncertainties) [38] is a unified robust algorithm that handles all types of uncertainty. The second approach is based on the Bayesian Stackelberg game model with dynamic execution uncertainty in which the uncertainty is represented using Markov Decision Process (MDP) where the time factor is incorporated.

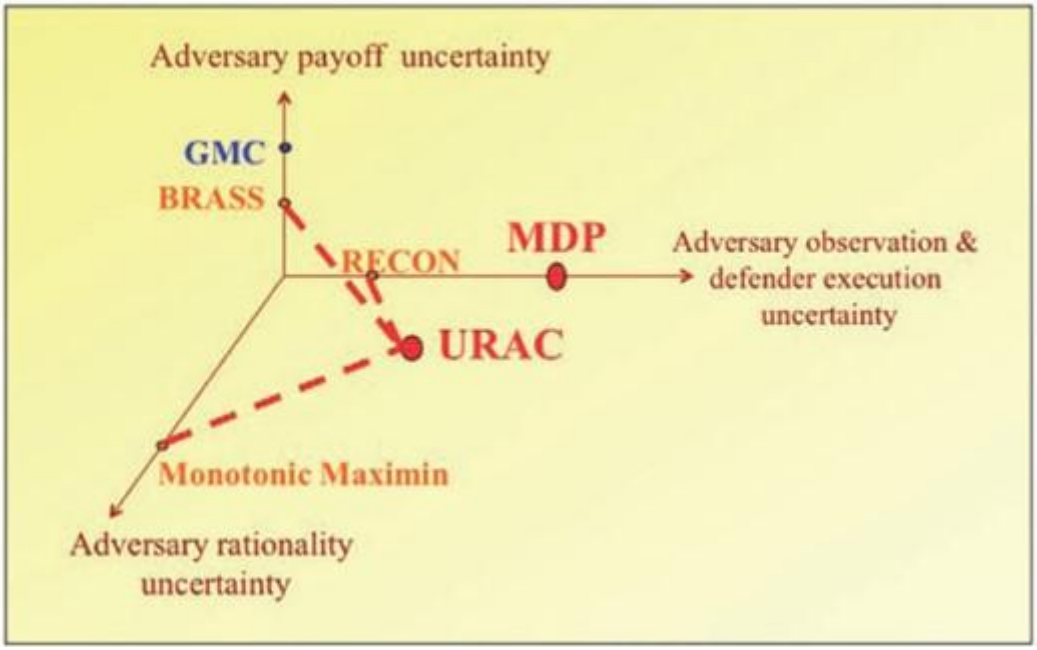


Figure 7.
Uncertainty space and algorithms.

In the following, we present two algorithmic solutions which are the representatives of these two approaches: U race — a unified robust algorithm to handle all types of uncertainty with uncertainty intervals and the MDP-based algorithm to handle execution uncertainty with an MDP representation of uncertainty.

Security patrolling with unified uncertainty space

Domain example—security in LAX

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

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

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: (in) placing vehicle checkpoints on inbound roads that service the LAX terminals, including both location and timing, and (ii) scheduling patrols for bomb-sniffing canine units at the different LAX terminals. The eight different terminals at LAX have very different 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 A rumor system focuses on two security measures at LAX (checkpoints and canine patrols) and optimizes security resource allocation using Bayesian Stackelberg games (Figure 8). Take the vehicle checkpoints model as an example. Assuming that there are *n*o roads, the police's strategy is placing $m < n$ checkpoints on these roads where *m* is the maximum number of checkpoints. Rumor randomizes allocation of checkpoints to roads. The adversary may conduct surveillance of this mixed strategy and may potentially choose to attack one of these roads. A rumor models different types of attackers with different payoff functions, representing different capabilities and preferences for the attacker. A rumor has been successfully deployed since August 2007 at LAX [1].

Figure 8.
LAX checkpoints are deployed using A rumor.

Although standard SSG-based solutions (i.e., D boss) have been demonstrated to improve the defender's patrolling effectiveness significantly, there remains potential improvements that can be made to further enhance the quality of such solutions such as taking uncertainties in payoff values, in the attacker's rationality, and in defender's execution into account. Therefore, we propose the unified robust algorithm, U race, to handle these types of uncertainties by maximizing the defender's utility against the worst-case scenario resulting from these uncertainties.

Algorithmic solution—uncertainty dimension reduction (U race)
In this section, we present the robust U race algorithm for addressing a combination of all uncertainty types [38]. 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, U race represents all these uncertainty types (except for the attacker's behaviors) using uncertainty intervals. Instead of knowing exactly values of these game attributes, the defender only has prior information with respect to the upper bounds and lower bounds of these attributes. For example, the attacker's reward if successfully attacking a target is known to lie within the interval.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

Furthermore, U race 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 QR model—a well-known human behavioral model for capturing the attacker's decision making [31].

Based on these uncertainty assumptions, U race tries 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 deciding the worst-case scenario. Nevertheless, U race introduces a unified representation of all these uncertainty types as an uncertainty set of attacker's responses. Intuitively, despite 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, U race 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 the traditional method is directly applied (i.e., taking the dual maximization of the inner minimization of maximin and merging it with the outer maximization—maximin now can be represented as a single maximization problem). Therefore, U race 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 can now be computed as a maximum over all the sub-maximin problems.

Security patrolling with dynamic execution uncertainty

Domain example T rusts for security in transit systems

Urban transit systems face multiple security challenges, including deterring fare evasion, suppressing crime and counterterrorism. 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. 9). 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 [39]. The 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).

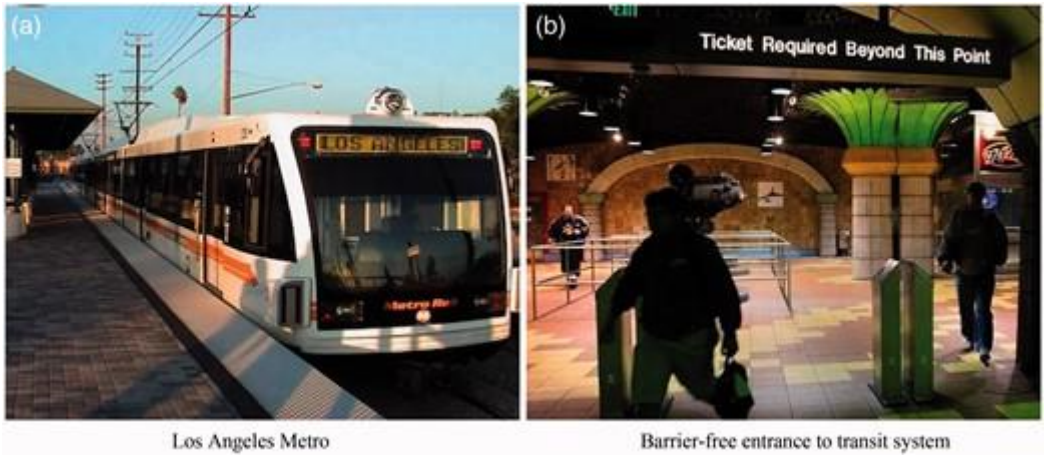


Figure 9.
Trusts for transit systems.

The T rusts model the patrolling problem as a leader–follower Stackelberg game [4]. The leader (LASD) recommits to a mixed strategy patrol (a probability distribution over all pure strategies), and riders observe this mixed strategy before

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

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, T rusts use a compact representation of the strategy space which captures the spatiotemporal structure of the domain.

The LASD conducted field tests of this T rusts 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.

Algorithmic solution—marginal MDP strategy representation

Utilizing techniques from planning under uncertainty (in particular, MDPs), we proposed a general approach to dynamic patrolling games in uncertain environments, which provides patrol strategies with contingency plans [40]. 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. We now describe the solution approach in more detail. Note that 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, because 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 T rusts carried out by 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 [40] 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 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 C ass 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. This interaction of the defender with the environment when executing patrol can be represented as an MDP.

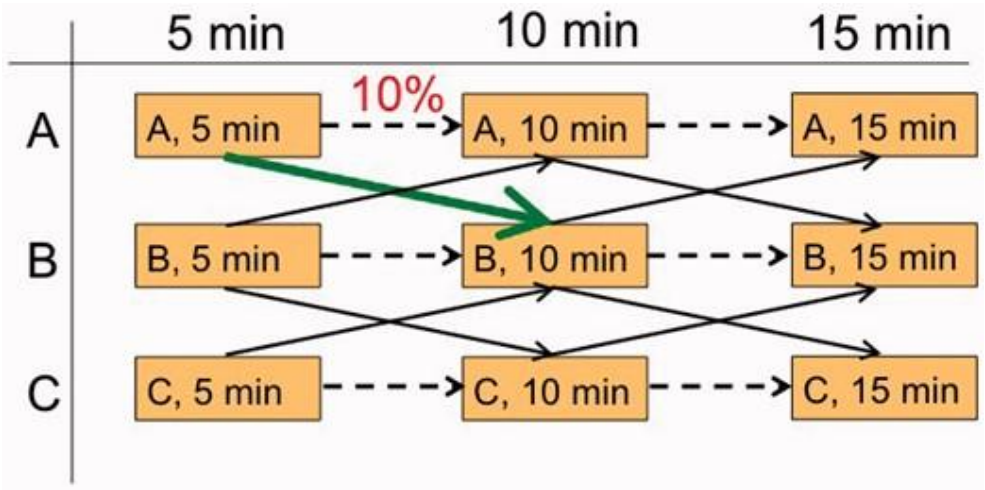


Figure 10.
An example of execution uncertainty in a transition graph.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

In essence, the transition graph as represented above is augmented to indicate the possibility that there are multiple uncertain outcomes possible from a given state. Solving this transition graph results in marginals over MDP policies. When a sample MDP policy is obtained and loaded on to a smart phone, it provides a patroller not only the current action but contingency actions also, should the current action fail or succeed. So, the MDP policy provides options for the patroller, allowing the system to handle execution uncertainty. 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. Therefore, instead of directly computing the mixed strategy, the defender attempts to compute the marginal probabilities of each patrolling unit reaching a state , and taking action *at* which have dimensions polynomial in the sizes of the MDPs (the details of this approach are provided in [40]).

Addressing bounded rationality in real-world problems

Game theory models the strategic interactions between multiple players who are assumed to be perfectly rational, i.e., they will always select the optimal strategy available to them. This assumption may be applicable for high-stakes security domains such as infrastructure protection where presumably the adversary will conduct careful surveillance and planning before attacking. However, there are other security domains where the adversary may not be perfectly rational due to short planning windows or because the adversary is less strategic due to lower stakes associated with attacking. Security strategies generated under the assumption of a perfectly rational adversary are not necessarily as effective as would be feasible against a less-than-optimal response. Therefore, addressing the boundedly rationality exhibited by human adversaries is a fundamental challenge for applying security games to wide variety of domains.

Domain example—green security domains

A number of our newer applications are focused on resource conservation, through suppression of environmental crime. One area is protecting forests [41], where we must protect a continuous forest area from extractors by patrols through the forest that seek to deter such extraction activity (Figure 11). 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 [41].

Figure 11.
Examples of illegal activities in green security domains.

Endangered species poaching is reaching critical levels as the populations of these species plummet to unsustainable numbers. The global tiger population, e.g., 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).

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

Behavioral modeling and learning

Recently, we have conducted some research on applying ideas from behavioral game theory (e.g., prospect theory [42] and QR [43]) within security game algorithms. One line of approaches is based on the QR model to predict the behaviors of the human adversary, and then to compute optimal defender strategies against such behavior of the adversary. These include BRQR [44] which follows the logit QR [43] model and subsequent work on SUQR models [32]. The parameters of these models are estimated by experimental tuning. Data from a large set of participants on the Amazon Mechanical Turk (AMT) were collected and used to learn the parameters of behavioral models to predict future attacks. In 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. 12) and deployed.

Figure 12.
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 [43] 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 [5]. Protection Assistant for Wildlife Security (PAWS) is an application which was originally created using Bayesian SUQR. However, in real-world security domains, we may

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

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 [21], which guarantees a bound for the loss of the defender to be within a constant factor of the adversary loss if the adversary responds nonoptimal. Another robust solution concept is monotonic maximin [37], 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 have been attempts to combine such robust optimization approaches with available behavior data [6] for RSSGs, resulting in a new human behavior model called Robust SUQR. However, one question of research is how these proposed models and algorithms will fare against human subjects in RSSGs. This has been explored in recent research [45] in the "first-of-its-kind" human subjects experiments in RSSGs over a period of 46 weeks with the "Wildlife Poaching" game, a brief summary of which is presented below.

In our human subjects experiments in RSSGs, we observe that: (I) existing approaches (QR, SUQR, Bayesian SUQR) [5 , 6 , 32] perform poorly in initial rounds, while Bayesian SUQR which is the basis for PAWS [5], perform poorly throughout all rounds; (ii) surprisingly, simpler models like SUQR which were originally proposed for single-shot games performed better than recent advances like Bayesian SUQR and Robust SUQR which are geared specifically toward addressing repeated SSGs. Therefore, we proposed a new model called SHARP (Stochastic Human behavior model with Attractiveness and Probability weighting) [45] which addresses the limitations of the existing models in the following way: (I) modeling the adversary's adaptive decision making process in repeated SSGs, SHARP reasons based on success or failure of the adversary's past actions on exposed portions of the attack surface, where attack surface is defined as the *n* -dimensional space of the features used to model adversary behavior; (ii) addressing limited exposure to significant portions of the attack surface in initial rounds, SHARP reasons about similarity between exposed and unexposed areas of the attack surface, and also incorporates a discounting parameter to mitigate adversary's lack of exposure to enough of the attack surface; (iii) addressing the limitation that existing models do not account for the adversary's weighting of probabilities, we incorporate a two-parameter probability weighting function. Based on our human subjects experiments highlighted in [45], we observe that SHARP completely outperforms existing approaches consistently over all rounds, most notably in initial rounds.

Addressing field evaluation in real-world problems

Evidence showing the benefits of the algorithms discussed in the earlier sections is definitely an important issue that is necessary for us to answer. Unlike conceptual ideas, where we can run thousands of careful simulations under controlled conditions, we cannot conduct such experiments in the real world with our deployed applications. Nor can we provide proof of 100% security—there is no such thing.

Instead, we focus on the specific question of: are our game-theoretic algorithms better at security resource optimization or security allocation than how they were allocated previously, which was typically relying on human schedulers or a simple dice roll for security scheduling (simple dice roll is often the other automation that is used or offered as an alternative to our methods). We have used the following methods to illustrate these ideas. These methods range from simulations to actual field tests.

1. Simulations (including using a machine learning attacker): we provide simulations of security schedules, e.g., randomized patrols, assignments, comparing our approach to earlier approaches based on techniques used by human schedulers. We have a machine learning-based attacker who learns any patterns and then chooses to attack the facility being protected. Game-theoretic schedulers are seen to perform significantly better in providing higher levels of protections [1 , 46]. This is also shown in Fig. 13 .
2. Human adversaries in the lab: we have worked with a large number of human subjects and security experts (security officials) to have them get through randomized security schedules, where some are schedules generated by our algorithms, and some are baseline approaches for comparison. Human subjects are paid money based on the reward they collect by successfully intruding through our security schedules; again our game-theoretic schedulers perform significantly better ([47]).
3. Actual security schedules before and after for some security applications, we have data on how scheduling was done by humans (before our algorithms were deployed) and how schedules are generated after the deployment of our algorithms. For measures of interest to security agencies, e.g., predictability in schedules, we can compare the actual human-generated schedules versus our algorithmic schedules. Again, game-theoretic schedulers are seen to perform significantly better by avoiding predictability and yet ensuring that more important targets are covered with higher frequency of patrols. Some of this data is published [2] and is also shown in Fig. 14 .

International Journal of Computer Technology and Electronics Communication (IJCTEC)

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

4. "Adversary" teams simulate attack: in some cases, security agencies have deployed adversary perspective teams or mock attacker teams that will attempt to conduct surveillance to plan attacks; this is done before and after our algorithms have been deployed to check which security deployments worked better. This was done by the USCG indicating that the game-theoretic scheduler provided higher levels of deterrence [2].
5. Real-time comparison human versus algorithm: this is a test we ran on the metro trains in Los Angeles. For a day of patrol scheduling, we provided head-to-head comparison of human schedulers trying to schedule 90 officers on patrols versus an automated game-theoretic scheduler. External evaluators then provided an evaluation of these patrols; the evaluators did not know who had generated each of the schedules. The results show that while human schedulers required significant effort even for generating one schedule (almost a day), and the game-theoretic scheduler ran quickly, the external evaluators rated the game-theoretic schedulers higher (with statistical significance) [48].
6. Actual data from deployment: this is another test run on the metro trains in Los Angeles. We had a comparison of game-theoretic scheduler versus an alternative (in this case a uniform random scheduler augmented with real-time human intelligence) to check fare evaders. In 21 days of patrols, the game-theoretic scheduler led to significantly higher numbers of fare evaders captured than the alternative [48 , 49].
7. Domain expert evaluation (internal and external): there have been of course significant numbers of evaluations done by domain experts comparing their own scheduling method with game-theoretic schedulers and repeatedly the game-theoretic schedulers have come out ahead. The fact that our software has now been in use for several years at several different important airports, ports, air-traffic, and so on, is an indicator to us that the domain experts must consider this software of some value.

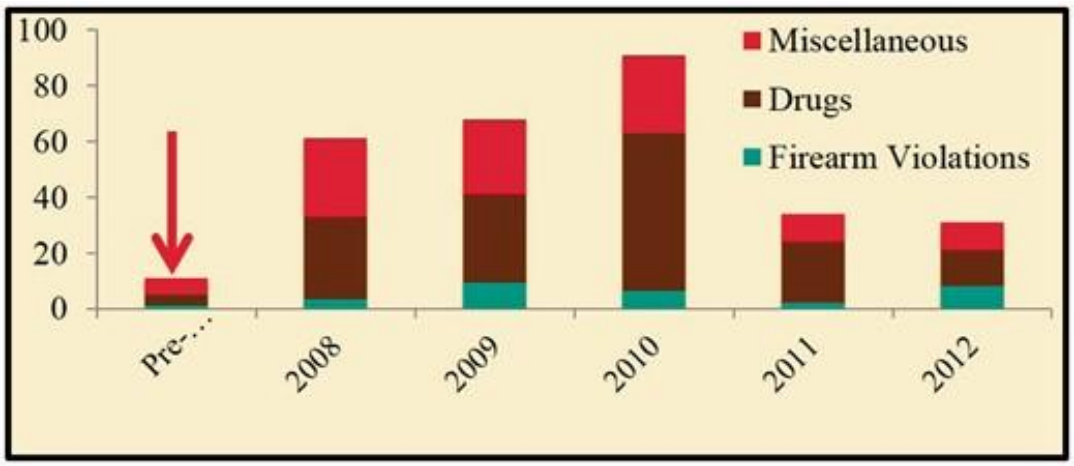
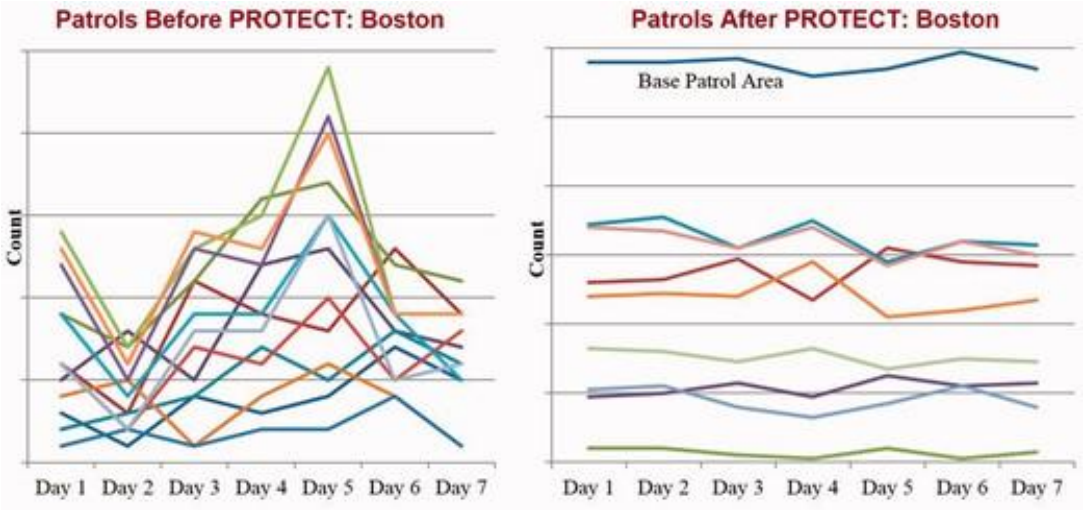


Figure 13.
A rumor evaluation results.



| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

Figure 14.
PROTECT evaluation results: redeployment (left) and post deployment patrols (right).

Cybersecurity: challenges and opportunities
The domain of computer security and privacy provides a rich set of challenges that requires new innovation and techniques. The application of game theory to cybersecurity is a new and promising research field. The potential benefits of applying game theory to cybersecurity problems are:
1. Game theory captures the adversarial nature of cybersecurity interactions and provides quantitative and analytical tools that may help find the optimal defense strategies.
2. Computer implementations of those methods allow examination of a large number of threat scenarios, which human analysts can miss due to cognitive limitations and biases.
3. Game theory provides methods for predicting actor's behavior in uncertain situations and suggesting probable actions along with predicted outcomes.

One of the challenges of applying game theory to cybersecurity is choosing the proper game model for a given security problem. Currently, selecting a game that has relevant features for being a cybersecurity scenario is primarily based on intuition. There is a lack of analysis and empirical data to confirm those choices. The problem is even more aggravated in the domain of privacy, where there is a fundamental tension between utility of data and privacy loss from data sharing.

Prior approaches based on simultaneous move games
Much work has focused on modeling cybersecurity problems as a simultaneous move game. The resulting Nash equilibrium analysis assume that all players: (I) form beliefs based on an analysis of what others might do (strategic thinking); (ii) choose the best response given those beliefs; and (iii) adjust best responses and beliefs until they are mutually consistent.

We survey some prior work on game theory approaches to cybersecurity problems. In the next section, we discuss security game-based approaches to cybersecurity problems. A popular model for the interaction between a defender and attacker in cybersecurity is the Flip It game [50]. This is a continuous time game that models the fact that any cybercaster will ultimately be compromised, and the defender will have to expend effort to detect and recover. Many variants of the game have been studied, such as playing Flip It with actual human subject experiments [51]. The Flip It model does not model the details of a defender–adversary interaction in a cyber-setting. As such, the model cannot be used directly in any real-world network to provide guidance about how to use and deploy defense resources at a fine-grained level.

There is lot of work on economics of security that uses game-theoretic reasoning. For example, Radonjic *et al.* [52] explores the economics and privacy aspects of internet service providers (ISPs) joining the online advertising market, analysis of security investments via insurance in different settings (weakest link, etc.) [53], and by considering interdependence and repeated interaction in context of security investments [54]. Recently, deception has been considered as a defense mechanism in network security. Pawlick *et al.* [55] consider a game model of honeypots used for network defense, where the interaction is modeled as a cheap talk game.

Although most of the existing literature on applications of game theory to cybersecurity assumes that the cost and effectiveness of the actions of the players are time independent, this is usually not true in practice. The cost and probability of success of attacks may vary over time: as much as the attacker's costs depend on the timing of the attack, the defender's costs depend on when to act to successfully thwart attacks. Recent work by Johnson *et al.* [56] has started to focus on the importance of analyzing the cost and effectiveness of players' actions in dynamic settings. They develop and analyze a continuous-time as well as a discrete-time model of the entire process occurring during the lifetime of a vulnerability, starting from its discovery till it is rendered useless and the on-going mitigation efforts by the defender while the vulnerability can still be exploited. Similarly, Rasouli *et al.* [57] has developed an analytical approach for dynamic cybersecurity problems that conure progressive attacks on a computer network. Importantly, the model recognizes the fact that the defender, i.e., the cyber-system supervisor, may not be able to observe the malicious actions of the adversary in real time. Thus, the defender must maintain a belief over the state of the computer network and include network monitoring as part of strategy in addition to deterring and mitigating cyber-attacks. Lu *et al.* [58] studies the active cyber defense in the setting of strategic attackers and/or strategic defenders. In particular, the paper combines control- and game-theoretic models under the "homogeneous" assumption that each compromised computer can attack the same part of computers. The work first studies two cases: infinite-time horizon best control and fast

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

optimal control when the attackers are nonstrategic. Then they provide the Nash equilibrium strategies in the case of strategic attackers.

A drawback of simultaneous move games with perfect rationality assumptions is the low predictability power of the model, as has been pointed out by behavioral economists [42 , 43]. The model places sufficient computational burden on the players, which results in the predicted outcomes not being realized in practice. Thus, often in economics such models are used for high level reasoning and post hoc understanding of the problem at hand. A day–to-day operational defense aid requires a detailed model of the problem as well as decent predictability.

Potential cybersecurity applications of security game techniques

Our Stackelberg game-based approach provides a complementary approach to the approaches described in the last subsection. In Stackelberg games the adversary's computational burden is lower and further we account for uncertainty and bounded rational behavior of the adversary. Even with a perfect rationality assumption, the overall lower complexity of computation of the Stackelberg equilibrium (as compared to the Polynomial Parity Arguments on Directed graphs (PPAD) complexity of Nash equilibrium) allows for a detailed model of the underlying domain with scalable algorithms to compute the equilibrium. There has been some initial use of the security games model to address problems in cybersecurity and privacy. We present three different potential applications here, two for cybersecurity and one for privacy policy enforcement in organization.

In [7], the authors study the problem of optimal resource allocation for packet selection and inspection to detect potential threats in large computer networks with multiple computers of differing importance. A number of intrusion detection and monitoring systems are deployed in real-world computer networks with the goal of detecting and preventing attacks. One countermeasure employed is to conduct "deep packet inspections," a method that periodically selects a subset of packets in a computer network for analysis but is costly in terms of throughput of the network. The security problem is formulated as a Stackelberg security game between two players: the attacker (or the intruder) and the defender (the detection system), which is played on a computer network modeled as a graph. The intruder wants to gain control over (or to disable) a valuable computer in the network by scanning the network, compromising a more vulnerable system, and/or gaining access to further devices on the computer network. The actions of the attacker can therefore be seen as sending malicious packets from a controlled computer (term source) to a single or multiple vulnerable computers (termed targets). The aim of the defender is to prevent the intruder from succeeding by selecting the packets for inspection, identifying the attacker, and after thwarting the attack. However, packet inspections cause unwanted latency and hence the defender has to decide where and how often to inspect network traffic in order to maximize the probability of a successful malicious packet detection. The authors provide polynomial time approximation algorithm that receives help from the sub modularity property of the discretized zero-sum variant of the game and finds solutions with bound error in polynomial time.

In a recent paper [8], the authors study the problem of optimal number of "honeypots" to be placed in a network. Honeypots are fake copies of electronic resources (servers, computers, routers, etc.) that aim to confuse the attacker so that the attacker attacks these honeypots. Attacks on honeypots also enable the defender to study the attacker and possibly catch them. The use of honeypots as a deceptive defense mechanism seems promising but has an associated cost in setting up the fake electronic assets. Thus, a central question in this defense mechanism is how many and which types of honeypots should be used? The authors use attack graphs to model possible attack trajectories that the attacker may use. The nodes in attack graphs are annotated with costs of attacks and benefits of successful attack, and also the probability of success of attack. In particular, the number and types of honeypots deployed influence the probability of success of attacks. The authors model the game as a Stackelberg security game with the defender choosing the number and type of honeypots to deploy. The attacker chooses an attack path with the best utility. The authors provide heuristic algorithms for the NP-hard problem of finding the optimal attack by converting the problem to an MDP.

One interesting work, called audit games [9 , 10], enhances the security games model with choice of punishments in order to capture scenarios of security and privacy policy enforcement in large organizations. Large organizations (such as Google, Facebook, and hospitals) hold enormous amounts of privacy sensitive data. These organizations mandate their employees to adhere to certain privacy policies when accessing data. Auditing of access logs is used by organizations to check for policy violating accesses and then the violators are punished. Auditing often requires human help to investigate suspicious cases and thereby arises the problem of allocating few resources to the vast number of cases to investigate. Another relevant question in this domain is how much should the organization punish in case of a violation? The audit game models the adversary as an agent that performs certain tasks (e.g., access to private data), and a subset of these tasks are policy violations. The auditor inspects a subset of the tasks and detects violations from

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

the inspected set. As punishments do affect the behavior of the adversary, it is critical for the auditor to choose the right level of punishment. Consequently, the choice of a punishment level is added to the action space of the auditor. However, punishment is not free for the auditor; the intuition being that a high punishment level creates a hostile work environment, leading to lack in productivity of employees which results in loss for the organization (auditor). Therefore, the auditor cannot impose infinite punishment and deter any adversary. The auditor's cost for a punishment level is modeled as a loss proportional to the choice of the punishment level. The auditor moves first by committing to an inspection and punishment strategy, followed by the best response of the adversary. The resultant Stackelberg equilibrium optimization turns out to be nonconvex due to the punishment variable. The authors present efficient algorithms for various types of scheduling constraints.

IV. CONCLUSION

Security is recognized as a world-wide challenge and game theory is an increasingly important paradigm for reasoning about complex security resource allocation. We have shown that the general model of security games is applicable (with appropriate variations) to varied security scenarios. There are applications deployed in the real world that have led to a measurable improvement in security. We presented approaches to address four significant challenges: scalability, uncertainty, bounded rationality, and field evaluation in security games. Cybersecurity provides added challenges that include limited observability and deception.

In short, we introduced specific techniques to manage each of these challenges. For scalability, we introduced three approaches: (I) incremental strategy generation for addressing the problem of large defender strategy spaces; (ii) double oracle incremental strategy generation with respect to large defender and attacker strategy spaces; (iii) compact representation of strategies for the case of mobile resources and moving targets; and (iv) cutting plane (incremental constraint generation) for handling multiple boundedly rational attacker. For handling uncertainty, we introduced two approaches: (I) dimensionality reduction in uncertainty space for addressing unification of uncertainties; and (ii) MDP with marginal strategy representation with respect to dynamic execution uncertainty. In terms of handling attacker-bounded rationality, we propose different behavioral models to capture the attackers' behaviors and introduce human subject experiments with game simulation to learn such behavioral models. Finally, for addressing field evaluation in real-world problems, we discussed two approaches: (I) data from deployment and (ii) mock attacker team.

While the deployed game-theoretic applications have provided a promising start, significant amount of research remains to be done. Cybersecurity provides challenges and opportunities in modeling multiple agents interacting in the extremely complicated cyber world. 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. Anek, O., Yin, Z., Jain, M., et al. (2012). 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). Richland, SC: IFAAMAS.
2. Durkota, K., Lisy, V., Kiekintveld, C., et al. (2015). Game-theoretic algorithms for best network security hardening using attack graphs. In Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS '15). Richland, SC: IFAAMAS.
3. Blocki, J., Christin, N., Datta, A., et al. (2013). Audit games. In Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI).
4. Blocki, J., Christin, N., Datta, A., et al. (2015). Audit games with multiple defender resources. In AAAI Conference on Artificial Intelligence (AAAI). Palo Alto, CA: AAAI Press.
5. von Stackelberg, H. (1934). Marktform und Gleichgewicht. Vienna: Springer.
6. Kiekintveld, C., Jain, M., Tsai, J., et al. (2009). Computing optimal randomized resource allocations for massive security games. In Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems (AAMAS), 689–696. Richland, SC: IFAAMAS.
7. Leitmann, G. (1978). On generalized Stackelberg strategies. Journal of Optimization Theory and Applications, 26, 637–643.
8. Navandar, Pavan. "Enhancing Cybersecurity in Airline Operations through ERP Integration: A Comprehensive Approach." Journal of Scientific and Engineering Research 5, no. 4 (2018): 457-462.

| ISSN: 2320-0081 | www.ijctece.com ||A Peer-Reviewed, Refereed and Bimonthly Journal |

|| Volume 5, Issue 5, September– October 2022 ||

DOI: 10.15680/IJCTECE.2022.0505003

9. Navandar, Pavan. " Enhancing Governance, Risk, and Compliance (GRC)" Journal of Scientific and Engineering Research 7, no. 3 (2020):250-256.
10. Navandar, Pavan. " Enhancing Security with Two-Factor Authentication in SAP Fiori Applications" Journal of Scientific and Engineering Research 5, no. 10 (2018):329-33.
11. Navandar, Pavan. " Segregation of Duties (SoD) Risks in SAP Security: Mitigation Strategies and Best Practices" Journal of Scientific and Engineering Research 6, no. 9 (2019):206-206.
12. Navandar, Pavan. " Unveiling the Power of Data Masking: Safeguarding Sensitive Information in the Digital Age" International Journal of Core Engineering & Management 5, no.6 (2019): 27-32.
13. Navandar, P. (2021). "Developing Advanced Fraud Prevention Techniques using Data Analytics and ERP Systems" Int J Sci Res, 10(5), 1326-1329.
14. Breton, M., Alg, A., & Haurie, A. (1988). Sequential Stackelberg equilibria in two-person games. Journal of Optimization Theory and Applications, 59, 71–97.
15. Conitzer, V., & Sandholm, T. (2006). Computing the optimal strategy to commit to. In Proceedings of the ACM Conference on Electronic Commerce (ACM-EC), 82–90.
16. Navandar, Pavan. " SAP Security is key for Business Success for ERP system" Journal of Scientific and Engineering Research 5, no. 6 (2018):398-400.
17. Navandar, P. (2021). Fortifying cybersecurity in Healthcare ERP systems: unveiling challenges, proposing solutions, and envisioning future perspectives. Int J Sci Res, 10(5), 1322-1325.
18. P. Navandar, "Optimizing SAP roles for efficient enterprise resource planning," Int. J. Sci. Res. (IJSR), vol. 9, no. 1, pp. 1932–1934, Jan. 2020, doi: 10.21275/SR24529194621.
19. P. Navandar, " Mitigating Financial Fraud in Retail through ERP System Controls" Int. J. Sci. Res. (IJSR), vol. 9, no. 4, pp. 1823–1827
20. Paruchuri, P., Pearce, J. P., Marecki, J., et al. (2008). 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), 895–902. Richland, SC: IFAAMAS.