7-2018

The price of usability: Designing operationalizable strategies for security games

Sara Marie MC CARTHY

Corine M. LAAN

Kai WANG

Phebe VAYANOS

Arunesh SINHA
Singapore Management University, aruneshs@smu.edu.sg

See next page for additional authors

Citation

MC CARTHY, Sara Marie; LAAN, Corine M.; WANG, Kai; VAYANOS, Phebe; SINHA, Arunesh; and TAMBE, Milind. The price of usability: Designing operationalizable strategies for security games. (2018). *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence (IJCAI-18),Stockholm, Sweden, July 13-19*. 454-460. Research Collection School Of Information Systems.
Available at: https://ink.library.smu.edu.sg/sis_research/4791

Author

Sara Marie MC CARTHY, Corine M. LAAN, Kai WANG, Phebe VAYANOS, Arunesh SINHA, and Milind TAMBE

The Price of Usability: Designing Operationalizable Strategies for Security Games

Sara Marie Mc Carthy
University of Southern California
sara.m.mccarthy@gmail.com

Corine M. Laan
University of Twente
c.m.laan@utwente.nl

Kai Wang
University of Southern California
wang319@usc.edu

Phebe Vayanos
University of Southern California
phebe.vayanos@usc.edu

Arunesh Sinha
University of Michigan
arunesh@umich.edu

Milind Tambe
University of Southern California
tambe@usc.edu

Abstract

We consider the problem of allocating scarce security resources among heterogeneous targets to thwart a possible attack. It is well known that deterministic solutions to this problem being highly predictable are severely suboptimal. To mitigate this predictability, the game-theoretic *security game* model was proposed which randomizes over pure (deterministic) strategies, causing confusion in the adversary. Unfortunately, such mixed strategies typically randomize over a large number of strategies, requiring security personnel to be familiar with numerous protocols, making them hard to operationalize. Motivated by these practical considerations, we propose an easy to use approach for computing strategies that are *easy to operationalize* and that bridge the gap between the static solution and the optimal mixed strategy. These strategies only randomize over an optimally chosen subset of pure strategies whose cardinality is selected by the defender, enabling them to conveniently tune the trade-off between ease of operationalization and efficiency using a single design parameter. We show that the problem of computing such operationalizable strategies is NP-hard, formulate it as a mixed-integer optimization problem, provide an algorithm for computing ϵ-optimal equilibria, and an efficient heuristic. We evaluate the performance of our approach on the problem of screening for threats at airport checkpoints and show that the *Price of Usability*, i.e., the loss in optimality to obtain a strategy that is easier to operationalize, is typically not high.

1 Introduction

The problem of protecting vulnerable targets from attackers using limited security resources manifests in many real world applications. A notable example, which serves as a key thrust in this paper, is the problem of screening for threats at (airport or border) checkpoints [AAA, 2014]. Deterministic solutions to these problems, which maintain the allocation of security resources constant, are highly predictable and thus severely suboptimal. The game-theoretic *security game* model was proposed as a means to mitigate this predictability against strategic adversaries [Tambe, 2011; Korzhyk *et al.*, 2010; Yin *et al.*, 2015; Balcan *et al.*, 2015; Basilico *et al.*, 2009]. Recently and in collaboration with the US Transportation Security Administration (TSA), this work was extended in the form of the *threat screening game* (TSG) model to tackle the problem of screening for threats at checkpoints. As this area of research continues to grow, security agencies have begun adopting these more sophisticated game-theoretic strategies making *ease of practical use and implementation* a key concern. Motivated by our discussions with the TSA, we focus our discussions in this paper on the problem of screening for threats.

There are two major issues that come in the way of practical operationalization of these game theoretic solutions to the problem of screening for threats at checkpoints.The first challenge relates to the practical difficulty of integrating decisions made by different levels of authority: *a)* higher level strategic decisions related to the design of teams of security resources and the assignment of personnel to shifts and teams; and *b)* lower level tactical decisions related to the online allocation of screenees to preformed teams of resources. Although these problems are intimately related, to date, no attempt has been made to address them in tandem making their solutions difficult to integrate and apply.

The second challenge relates to the fact that optimal solutions to TSGs typically involve randomizing over large numbers of pure strategies, each corresponding to a different security protocol. Thus, while they are by far preferable to deterministic strategies from an efficiency perspective, they are difficult to operationalize, requiring the security personnel to be familiar with numerous protocols in order to execute them.

Contributions We address these two shortcomings related to usability of TSG. First, we build upon the work on *Simultaneous Optimization of Resource Teams and Tactics* (SORT) [Mc Carthy *et al.*, 2016] to propose SORT-TSG, a model for TSG that yields solutions that can be directly applied in their entirety in practice and where security personnel schedules are integrated with screening needs. Second, we propose an easy to use mixed-integer optimization model for computing strategies that are *easy to operationalize* and that bridge the gap between the suboptimal deterministic solution and the optimal yet impracticable mixed strategy. These strategies only randomize over an optimally chosen subset of pure strategies whose cardinality is selected by the defender, enabling them to conveniently tune the trade-off between ease of operationalization and efficiency using a single design parameter. The optimization formulation does not scale to realistic size instances and we propose a novel solution approach for computing ϵ-optimal equilibria as well as a heuristic for computing operationalizable strategies to SORT-TSG. We perform extensive numerical evaluation that showcases the solution quality and scalability of our approach and illustrate that the Price of Usability is typically not high.

2 Usability in Security Games

Security games deal with the challenge of allocating scarce security resources among heterogeneous targets to avert a possible attack. These problems can be formulated as Stackelberg games between the defender (leader) and the attacker (follower) and admit an optimal (albeit challenging to compute) mixed strategy solution.

Definition 1 (Mixed Strategy in a Security Game)**.** *The defender pure strategies are given by a finite set of integral points $\mathcal{Q} \subset \mathbb{N}^n$ where n denotes the number of targets and $S := |\mathcal{Q}|$ is the number of pure strategies. Intuitively, each $q \in \mathcal{Q}$ corresponds to a (static) allocation of security resources. To maximize the chances of thwarting an attacker, the defender randomizes over pure strategies to build a mixed strategy, defined as a distribution over pure strategies:*

$$\mathcal{P} := \left\{ p \in \mathbb{R}^n \ : \ p_i \geq 0, \ i = 1, ..., S, \ \sum_{i=1}^{S} p_i = 1 \right\}.$$

The objective of the defender is to select the mixed strategy that maximizes her expected utility assuming best response from the adversary. For our focus domain of threat screening at airports [Brown *et al.*, 2016; Schlenker *et al.*, 2016; McCarthy *et al.*, 2017], the corresponding TSG model often has an extremely large pure strategy space. Further, the number of pure strategies in the support of a mixed strategy is also large as we show in our experiments.

In practice, each pure strategy can be viewed as a separate *security protocol*. In the context of TSG, these are different configurations and number of screening equipment. Thus, mixed strategies with large support sets can be problematic to operationalize as they require security agents to be familiar with a large variety of protocols to execute them all properly. These types of complex tasks increase the cognitive load in individuals [Hogg, 2007] increasing the likelihood that mistakes are made [Paas, 1992; Cooper and Sweller, 1987] and making the system vulnerable to exploitation. While such usability concerns have always been present in deployed security games, these have often been addressed in an ad-hoc fashion, and not explicitly discussed in the literature. For example, the US Coast Guard limited the number of pure strategies used in the Staten Island Ferry security game to avoid cognitive overload for boat operators [Fang *et al.*, 2013][Fang private communication 2018]. To the best of our knowledge, [Paruchuri *et al.*, 2007] is the only paper that explicitly discussed limiting the number of pure strategies in security games; although they only handled small games (100s pure strategies), and they did not consider the impact of such restriction on solution quality. In this paper, we explicitly formulate usability in security games and motivate our definition to limit the cognitive load placed on security personel, and refer to such strategies as being *operationalizable*.

Operationalizable Security Games Motivated by our discussions with practitioners in the security domain, we propose a model for usability in security games which we refer to as *Operationalizable Security Games* that admits solutions whose mixed strategy support cardinality is a design parameter selected by the defender; the choice of cardinality enables explicit trade-off between ease of implementation and efficiency. Rather than pre-committing to a fixed subset of pure strategies to be used in the randomization, our model decides on the *best* subset of policies to employ. Our hope (which we confirm with extensive experiments, see Section 5) is that the *price of usability*, i.e., the loss in efficiency due the restriction of the space of feasible mixed strategies, will not be high even if only a moderate number of strategies is employed.

Definition 2 (k-Operationalizable Mixed Strategy)**.** *A mixed strategy p is k-operationalizable if the cardinality of the support of p is limited to k, i.e.* $|\{i \in \{1, \ldots, S\} : p_i > 0\}| \leq k$.

For usability's sake, we propose to restrict solutions of security games to be k-operationalizable. A large k produces solutions that randomizes over a large number of pure strategies (maximizes optimal utility but not easy to operationalize) and low k produces more deterministic strategies (easy to operationalize, but exploitable by an intelligent adversary). We can balance between usability and efficiency using the single parameter k. The following theorem postulates that unfortunately usability comes at a *computational price*.

Theorem 1. *Let $\mathcal{G}$ be a zero sum game with pure strategy space $\mathcal{Q}$. The problem of finding optimal solutions that are k-operationalizable is NP-Hard to solve even if $\mathcal{G}$ can be solved in polynomial time.* [1]

For a player in such a game $\mathcal{G}$, an optimal mixed strategy which is k-operationalizable is one which minimizes that player's *Price of Usability.*

Definition 3 (Price of Usability)**.** *Let G be a game with optimal mixed strategy solution p and utility $U(p)$. Let p^k be an optimal k-operationalizable mixed strategy solution to G. We define the price of usability (PoU) as the ratio between the utilities of p and p^k so that $PoU := {}^{U(p)}/_{U(p^k)}$.*

3 SORT for Threat Screening Games

Addressing the two usability limitations of TSG [Brown *et al.*, 2016; Schlenker *et al.*,] discussed earlier, in this section we first present (1) the model of Simultaneous Optimization (SORT) for TSGs and second (2) the problem of computing operationalizable strategies for TSGs. We assume a zero-sum game. Throughout this section we use the example of passenger screening at airports, but emphasize that the TSG applies to generalized screening problems.

[1] All proofs can be found in the appendix at: http://teamcore.usc.edu/papers/2018/POU_appendix.pdf

3.1 Problem Description

A TSG is a game between the screener (defender) and adversary over a finite planning horizon $\mathcal{W}$ consisting of W time windows. The defender is operating a checkpoint through which screenees (passengers) arrive at during each time window. Each screenee belongs to a category $c \in \mathcal{C}$ where a category $c := (\rho, f)$ consists of components which are controllable and uncontrollable. In the airport security domain, the controllable component f corresponds to a flight type, dictated by the adversary's choice of flight to attack, while the uncontrollable element ρ describes the risk level assigned to each passenger (i.e. if they are TSA pre-check). It is assumed that the number of passengers of category c arriving during each time window, N_c^w, is known.

The adversary attempts to pass through screening by disguising themselves as one of the screenees. He has a choice of flight to attack, and thus can choose his flight type category, a time window w to arrive in and an attack method $m \in \mathcal{M}$. The adversary cannot control his risk level ρ and we assume a prior distribution P_ρ over the risk level of the adversaries.

At the checkpoint, the defender has a set of $r \in \mathcal{R}$ resources which are combined into teams indexed in the set $\mathcal{T}$ to which incoming passengers are assigned. If a passenger is assigned to be screened by a team $t \in \mathcal{T}$, they must be screened by all resources $\mathcal{R}(t) \subset \mathcal{R}$ in that team. The efficiency of a team, $E_{t,m}$, denotes the probability that an attacker carrying out an attack of type m be detected when screened by team t. This efficiency depends on the resources in that team: $E_{t,m} = 1 - \prod_{r \in \mathcal{R}}(1 - e_{r,m})$, where $e_{r,m}$ is the efficiency of resource r against attack method m.

Each resource $r \in \mathcal{R}$ has a fixed capacity C_r for the number of passenger which it can process in any time window. In the case that it is not possible to screen all passengers in a single time window, we allow these passengers to be screened in the next time window by their assigned resources, at a cost ϕ_r per passenger overflowing to the next window. Each resource r maintains an overflow queue o_r^w corresponding to the number of passengers waiting to be processed by that resource in the beginning of time window w.

To speed up checkpoint processing, the defender can increase the number of resources of each type that are available in a particular window (by e.g., opening up more lanes). However, the number of resources of each type r that can be operated at any given time is limited by the number of resources of that type that are available in the arsenal of the defender, denoted by $M_r \in \mathbb{R}$, and by the number of operators that are working in that window. Specifically, to operate each resource of type r, A_r workers are needed. The workforce of the defender consists of S workers and the defender can decide on the number of workers available in any window. However, the workers must follow shifts: they can start in arbitrary time windows but must work for δ consecutive time windows.

3.2 SORT-TSG Problem Formulation

We now formulate the SORT problem for TSG as a mixed-integer linear optimization problem. For convenience, we first introduce the pure strategy spaces related to the strategic and tactical decisions of the defender, respectively, and then go on to formulate the optimization problem which randomizes over these strategies.

The core strategic decisions of the SORT-TSG problem correspond to the number of resources of each type $r \in \mathcal{R}$ to operate in each window, which we denote by $y_r^w \in \mathbb{N}^+$. They also include the number of workers $b^w \in \mathbb{N}^+$ to start their shift in window w and the number of workers s^w available in window w. The space of pure strategic strategies can then be expressed as:

$$\mathcal{Y} := \left\{ y : \exists(b,s) : \begin{array}{ll} s^w = \displaystyle\sum_{w'=\max(1,w-\delta+1)}^{\min(w,W-\delta+1)} b^{w'} & \forall\, w \\ \displaystyle\sum_{w=1}^{W-\delta+1} b^w \leq S & \\ \displaystyle\sum_{r \in \mathcal{R}} y_r^w A_r \leq s^w & \forall\, r \\ y_r^w \leq M_r & \forall\, w, r \\ y_r^w, b^w, s^w \in \mathbb{N}^+ & \forall\, w, r \end{array} \right\}.$$

The first constraint above counts the total number of workers with shifts currently in progress at time window w. The second constraint stipulates that the total number of workers assigned to each shift cannot exceed the size of the workforce. The third and fourth constraints enforces that in each time window there must be enough workers to operate each resource, and that the number of operating resources cannot exceed the maximum number available for each type.

The core tactical decision variables of the SORT-TSG problem correspond to the number of passengers of each type c to screen with team t in window w, denoted by $n_{t,c}^w$. For any choice y of strategic decisions, the space of pure tactical strategies is expressible as:

$$\mathcal{X}_y := \left\{ (n,o) : \begin{array}{ll} \displaystyle\sum_{t \in \mathcal{T}} n_{c,t}^w = N_c^w & \forall\, c, w \\ \displaystyle\sum_{t : r \in R(t)} \sum_{c} n_{c,t}^w \leq y_r^w C_r - o_r^{w-1} + o_r^w & \forall\, w, r \\ n_{t,c}^w, o_r^w \in \mathbb{N}^+ & \forall\, t, c, w, r \end{array} \right\},$$

where the two constraints above stipulate that all arriving passengers must be assigned to be screened by a team and enforce the capacity constraints on each of the resource types. Note that the capacity is determined by the number of operating resources of each type. The full defender pure strategy space can be expressed compactly as:

$$\mathcal{Q} := \{(y,n,o) : y \in \mathcal{Y},\ (n,o) \in \mathcal{X}_y\}.$$

Next, given the probability distribution as the defender's mixed strategy, we denote by $E_p[\ ]$ the expectation operator with respect to p (the mixed strategy). Thus, the expected number $n_{t,c}^w$ of passengers in category c screened by team t in time window w and the expected number o_r^w of passengers waiting to be screened by a resource of type r in time window w are given by:

$$E_p[n_{t,c}^w] := \sum_{i=1}^{S} p_i n_{t,c}^{w,i} \quad \text{and} \quad E_p[o_r^w] := \sum_{i=1}^{S} p_i o_r^{w,i}. \qquad (1)$$

The utility of the defender is linear in the pure strategies, so the defender's optimization problem can be expressed as:

$$\begin{array}{lll} \max\limits_{p} & \sum_\rho P_\rho\theta_\rho - \sum_w \sum_r \phi_r E_p[o_r^w] & \\ s.t. & \theta_\rho \le z^w_{c,m}U_c^+ + (1-z^w_{c,m})U_c^- & \forall\, c,m,w \\ & z^w_{c,m} = \sum_t E_{t,m}\frac{E_p[n^w_{c,t}]}{N_c^w} & \forall\, c,m,w \\ & p\in\mathcal{P}, & \end{array} \qquad (\mathscr{P})$$

where $z^w_{c,m}$ is the adversary's detection probability for an adversary of type c, using attack m during w and θ_ρ is the expected utility when the passenger's risk level is ρ. We denote this formulation of the SORT-TSG as problem $\mathscr{P}$.

Theorem 2. *Problem $\mathscr{P}$ is NP-Hard to solve.*

Reduces to [Brown *et al.*, 2016] when there is no overflow and when strategic decisions are fixed.

3.3 Operationalizable Strategies for SORT-TSG

The SORT-TSG problem admits additional usability concerns; not only can the mixed strategy have a very large support, but the number of types of resource configurations (teams) used by any pure strategy may also be very large (as the number of team types grows combinatorially with the number of resources). This can also pose the same operationalization issues, and so we also propose to limit the number of possible resource configurations that may be used in any pure strategy. Formally, we say that a mixed strategy solution to a SORT problem is operationalizable if the following property holds.

Definition 4 ((k,τ)-Operationalizable Mixed Strategy)**.** *A mixed strategy p is said to be (k,τ)-operationalizable if the support size of p is less than k, and each pure strategy uses no more than τ unique teams, i.e., if l_t is a binary variable indicating the formation of a team of type t then $\sum_{t=1}^{T} l_t \le \tau$.*

We can compute *operationalizable strategies* for the TSG problem by constructing a new set of allowed pure strategies $\mathcal{Q}_\tau$ by adding the following additional constraints to the set Q which enforce that each pure strategy may use no more than than τ resource configurations:

$$\sum_{t=1}^{T} l_t \le \tau; \quad \frac{n^w_{c,t}}{N^w_c} \le l_t, \ \ \forall\, t,w,c; \quad l_t \in \{0,1\}, \ \ \forall\, t. \quad (2)$$

Where the second constraint enforces that $l_t = 1$ if a strategy uses team t at any point, i.e., if $\exists\ w,c,t\ :\ n^w_{c,t} > 0$ We then enforce that the support of the mixed strategy has maximum cardinality k by replacing equations (1) with:

$$E_p[n^w_{t,c}] = \sum_{i=1}^{k} p_i n^{w,i}_{t,c} \quad E_p[o^w_r] = \sum_{i=1}^{k} p_i o^{w,i}_r \quad \forall\, w,t,c, \quad (3)$$

such that $p \in \mathcal{P}_k = \{p_i \ge 0, i = 1,...,k,\ \sum_{i=1}^{k} p_i = 1\}$. Lastly, for the TSG problem it is undesirable to have many different schedules for staff members and have employees work different shifts throughout the week. For this reason we specifically enforce that the scheduling decisions s should be the same across all k pure strategies i.e.

$$s_i = s_j \ \ \forall i,j \in \{0\ldots k\}. \quad (4)$$

These additions (2,3,4) to $\mathscr{P}$, define the operationalizable SORT-TSG problem, we refer to the problem as $\mathscr{P}$k .

4 Solving the Operationalizable SORT-TSG

The SORT-TSG problem is expressible as a mixed integer linear program (MILP). However the resulting operationalizable problem $\mathscr{P}$k is non-linear, with bilinear terms introduced in (3). Since the domains of n and o are finite we can express each integer variable n and o as a sum of binary variables, and the bilinear terms can be easily linearized using standard optimization techniques. However, the resulting program has a number of binary variables which grows with the number of passengers, making the full MILP formulation intractable to solve. Other standard approaches for dealing with these types of problems, such as column generation, also do not work well as we show in Section 5. In the following, we provide our new solution approach for efficiently solving $\mathscr{P}$k .

For convenience we define the following notation. Let $\mathscr{P}$ be an optimization problem with integer variables $x_i \in \mathbb{N}\ \ \forall i$. We denote the LP relaxation of $\mathscr{P}$, i.e., the problem obtained by letting $x_i \in \mathbb{R}\ \ \forall i$, as $\mathscr{P}^{\,LP}$. Additionally let the LP relaxation of a problem $\mathscr{P}$ with respect to a single variable x_j, i.e., the problem obtained by letting $x_j \in \mathbb{R}$, be denoted by$\mathscr{P}^{\,LP_{x_j}}$. Let the marginal value of x_j (i.e., the expectation $E_p[x_j]$) be denoted $\tilde{x}_j$. Lastly we denote the problem with a fixed variable x_j as $\mathscr{P}\,|x_j$.

Our novel solution approach $\mathscr{P}$k is based on the two following ideas: (1) we allow the k pure strategies to form a *multiset* (so that a single strategy may appear multiple times) and (2) we restrict the mixed strategy to be a *uniform distribution* over the multiset of k pure strategies. The intuition behind this approach is that the multiset allows us to approximate any optimal mixed strategy $\mathcal{P}$ using multiples of the fractions $\frac{1}{k}$. If $p_i \ge \frac{1}{k}$ (probability of playing strategy i), then strategy i will appear multiple times in the multiset, and thus will be played with probability $\frac{a}{k}$ where a is the number of times it appears. If $p_i < \frac{1}{k}$ then as k grows large enough, the loss in utility from not playing strategy i becomes negligible.

This intuition is formalized in Theorem 3 which stipulates that we can compute approximate equilibria (with approximation error ϵ) for any choice of k by fixing a uniform distribution over the multiset of k pure strategies.

Theorem 3. *Given a game G with Stackelberg equilibrium x^*, z^* and game value $(x^*)^\top Rz^*$ there exists a solution x', z' such that x' is k-operationalizable and is* uniformly distributed over its support *where for $k > \frac{4\log(1+n)}{\epsilon^2}$ (where n is the size of the adversary's action space) we have that x', z' is an ϵ-Stackelberg equilibrium with game value $(x^*)^\top Rz^* - (x')^\top Rz' \le \epsilon$.*

We derive these bounds following the proof of [Lipton *et al.*, 2003], which for our problem are a factor 3 tighter. By fixing $p = \frac{1}{k}$, $\mathscr{P}$k can be solved directly as an MILP without the creation of extra binary variables. Algorithm 1 outlines this process. To speed up computation we first solve the full relaxation $\mathscr{P}\text{k}^{\,LP}$ to get marginal values $\tilde{y}$ and $\tilde{n}$ (line 2). We then round these to get integral values y^r and n^r (line 3) which we then use as a warm start to solve the MILP (line 5).

For any choice of k, we can then compute an ϵ-equilibrium and show that in practice this approach performs well. Additionally, it provides a general framework from which we can

Algorithm 1 k-Uniform Strategies

```
1: procedure k-UNIFORM
2:     $\tilde{y}, \tilde{n}, \tilde{o} \leftarrow \mathscr{P}\text{k}^{\,LP}$
3:     $y^r, n^r, o^r \leftarrow$ Round($\tilde{y}, \tilde{n}, \tilde{o}$)
4:     $p \leftarrow p_i = \frac{1}{k},\ i = 1, ..., k$
5:     $y, n, o \leftarrow$ WarmStart($\mathscr{P}\text{k}\ |q_k, y^r, n^r, o^r$)
6:     return $y, n, o$
```

build more sophisticated and scalable algorithms which we demonstrate in the next section.

4.1 Heuristic Approach

While the approach described in Section 4 provides guarantees for any choice of k, in practice the problem can still be slow to solve, as it requires solving an MILP. Thus we provide a heuristic approach which can be solved more efficiently and still yields high solution quality in practice.

The novelty in our approach comes from exploiting the hierarchical structure of the SORT variables, as well as an optimized rounding procedure to decompose marginal solutions into an operationalizable set of k integral strategies.

The tactical variables (n, o) are dependent on the strategic variables y and so, starting from marginal solution to the LP relaxation, we first impose the operationalizable constraints on the strategic variables, keeping the tactical variables unconstrained. This gives us a set of k strategies with integral y, from which we can compute the corresponding integral tactical variables n for each of the k strategies. Both of these steps use an *optimized rounding procedure*. Because our objective is a function of the expected value of n and o, it becomes important to optimize over how we do our rounding. Ideally we would like to be able to exactly reconstruct the marginal values obtained from the LP relaxation in order to maximize our objective. Arbitrarily rounding the marginal variables to generate k integral strategies does not take into account the value of the resulting marginal and may result in very suboptimal solutions. Instead we compute an optimal rounding to compute feasible solutions, which take into account the value of the resulting marginal with respect to our objective.

Algorithm 2 outlines the steps of this solution method. We start by solving the full relaxation $\mathscr{P}\text{k}^{\,LP}$ (line 2) to obtain a marginal solution for the strategic variables $\tilde{y}$. We then decompose this marginal solution into a set of k integral pure strategies (line 3) using an optimized rounding procedure (which we formalize in the later section) which computes the best k roundings of the marginal $\tilde{y}$ (keeping a marginal $\tilde{n}_i$ for each strategy i, $i = 1, ..., k$). We then compute the best integral assignment n_i and corresponding overflow o_i for each resource configuration y_i (line 4) using the same optimized rounding procedure on the marginals $\tilde{n}_i$, $i = 1, ..., k$.

Strategic Variables: Resource Configurations At this stage (line 3) we determine what the k optimal integral variables y_i are assuming no integrality constraints on the n_i variables, i.e. we solve the problem $\mathscr{P}\text{k}^{\,LP_n}$ equivalent to letting $E_p[n^w_{t,c}] = \sum_{i=1}^k p_i \tilde{n}^{w,i}_{t,c}$ and $E_p[o^w_r] = \sum_{i=1}^k p_i \tilde{o}^{w,i}_r$, where $\tilde{n}^{w,i}_{t,c}, \tilde{o}^{w,i}_r$ are in the integer relaxation of $\mathcal{X}_{y_i}$. Unfortunately the following theorem shows that $\mathscr{P}\text{k}^{\,LP_n}$ is still intractable to solve.

Algorithm 2 Multiple Hierarchical Relaxations

```
1: procedure MHR
2:     $\tilde{y}, \tilde{n}, \tilde{o} \leftarrow \mathscr{P}\text{k}^{\,LP}$
3:     $y_i, \tilde{n}_i, \tilde{o}_i\ \ i = 1, ..., k \leftarrow$ Strategic($\mathscr{P}\text{k}^{\,LP_n}|q, \tilde{y}$)
4:     $y_i, n_i, o_i\ \ i = 1, ..., k \leftarrow$ Tactic($\mathscr{P}\text{k}\ |y, \tilde{n}$)
5:     return $y, n, o$
```

Theorem 4. *Problem $\mathscr{P}\text{k}^{\,LP_n}$ is NP-hard to solve.*

To approximate this problem, as in Algorithm 1, we assume a uniform distribution for the mixed strategy. Note that by Theorem 3 for any choice of k that $\mathscr{P}\text{k}^{\,LP_n}|p$ is an ϵ approximation to $\mathscr{P}\text{k}^{\,LP_n}$. Given $\mathscr{P}\text{k}^{\,LP_n}|p$, we compute a multiset of k integral solutions y_i, $i = 1, ..., k$, from the marginal $\tilde{y}$ using the following optimized rounding procedure. We make the change of variables $y_i = \lfloor \tilde{y} \rfloor + \delta_i$ such that $\delta_i \in \mathbb{N}^+$, $i = 1, ..., k$. Solving $\mathscr{P}\text{k}^{\,LP_n}|p$ with this change of variables computes the best k roundings of the marginal $\tilde{y}$ which we use as our k pure strategies. This subroutine is outlined in Algorithm 3.

Algorithm 3 Determine resource allocations y_s

```
1: procedure STRATEGIC($\mathscr{P}\text{k}^{\,LP_n}|p$, $\tilde{y}$)
2:     $p \leftarrow p_i = \frac{1}{k},\ i = 1, ..., k$
3:     $y \leftarrow y_i = \lfloor \tilde{y} \rfloor + \delta_i,\ \delta_i \in \mathbb{N}^+,\ i = 1, ..., k$
4:     return $\underset{y, \tilde{n}, \tilde{o}}{\text{argmax}}\ \mathscr{P}\text{k}^{\,LP_n}|p$
```

Tactic Variables: Passenger Allocations (line 4) We now have for each pure strategy i, a marginal $\tilde{n}_i$, $i = 1, ..., k$. In this step we again apply the same optimized rounding procedure to these variables to obtain integral values n_i. Additionally, here we relax the constraint $p_i = \frac{1}{k}$ and allow the program to optimize over the distribution over pure strategies.

Reintroducing the mixed strategy p as a variable reintroduces the bilinear terms (3) in $\mathscr{P}\text{k}$. However, with our rounding procedure, we can efficiently linearize these terms without creating a very large number of binary variables (as with the full MILP). We let $n_i = \lfloor \tilde{n}_i \rfloor + \gamma_i$ and are left with the bilinear terms $p_i(\gamma_i)$. To linearize these we make a change of variable $z_i = p_i(\gamma_i)$ and can express constraints (3) as:

$$\begin{aligned} E[n^w_{t,c}] &= \sum_{i=1}^{k} \left(p_i \left\lfloor \tilde{n}^{w,i}_{t,c} \right\rfloor + z^{w,i}_{t,c} \right), \\ 0 \le z^{w,i}_{t,c} &\le p_i, && \forall\, i = 1, ..., k. \\ z^{w,i}_{t,c} &\ge p_i - (1 - \gamma^{w,i}_{t,c}), && \forall\, i = 1, ..., k. \end{aligned}$$

This subroutine is outlined in Algorithm 4. First, we make the change of variable for the rounding procedure, and linearize the bilinear terms. We then solve the resulting optimization problem for the fixed y and b solved in the previous stage of the algorithm and finally return n and p which gives us a complete (k, τ)-operationalizable solution.

Algorithm 4 Determing Passenger type allocation n_s

```
procedure TACTIC($\mathscr{P}$k $|(y,b)$, $\tilde{n}$)
    $n \leftarrow n_i = \lfloor \tilde{n}_i \rfloor + \delta_{n_i},\ \delta_{n_i} \in \mathbb{N}^+, i = 1, ...k$
    LinearizeTerms($\mathscr{P}$k $|(y,b)$)
    return $\operatorname{argmax}_{n,p} \mathscr{P}$k $|(y,b)$
```

5 Evaluation

We evaluate our algorithms on several different sized instances of SORT-TSG. We use instances of three types: small, moderate and large instance with time windows, passenger types and resources (W=1, C=2, R=2), (W=5, C=10, R=5), (W=10, C=20, R=5) respectively. The large instances correspond to a 10 hour planning window for a single terminal at a large airport. [2] Each experiment is averaged over 50 randomized instances of the remaining parameters.

The Price of Usability In this paper, we proposed to mitigate the *price of usability (PoU)* by computing (k, τ)-operationalizable strategies. We have defined the price of usability similarly to the price of anarchy, as the ratio between the optimal solution with no usability constraints and the operationalizable equilibrium, (i.e. $\mathscr{P}$ /$\mathscr{P}$k) so that when the operationalizable game $\mathscr{P}$k has the same optimal objective as $\mathscr{P}$ the $PoU = 1$. In order to compare the operationalizable utility to that of $\mathscr{P}$, we use column generation to compute the optimal solution to the security game without usability constraints. We do this for moderately sized games, as the column generation method does not scale up to large instances. In Figure 1, we show that the PoU shrinks to almost 1 with increasing number of pure strategies k and team types τ. We note that the bump in runtime with increasing τ is due to a phenomenon in security games known as the deployment to saturation ratio [Jain *et al.*, 2012].

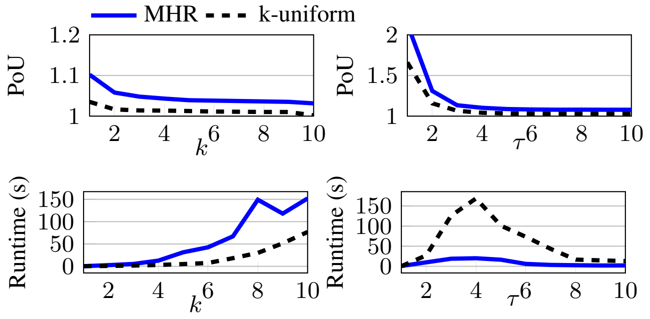

Figure 1: Here we show the empirical PoU, as well as the runtimes of both methods with increasing k and τ for both methods (left: $\tau = 10$, right: $k = 2$).

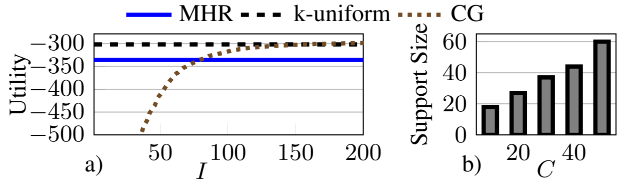

Figure 2: a) Comparison of our algorithms with CG which is cut off after I iterations ($k = 5$, $\tau = 10$). b) Support size of CG solutions for increasing problem size.

Solution quality We evaluate the solution quality of our algorithms by comparing to (1) two variations of a column generation heuristic, one which cuts off after I iterations and one which selects and re-optimizes over the top k strategies, and (2) the full MIP which optimally solves operationalizable security game $\mathscr{P}$k . Figure 2(a) shows the comparison of our methods with the first column generation (CG) baseline. When run to convergence, (CG) optimally solves $\mathscr{P}$, without operationalizable constraints (CG). We approximate $\mathscr{P}$k by cutting off (CG) after I iterations. We see that for small I, CG achieves very poor utilities compared to our algorithm, and that it takes up to 150 generated strategies (iterations) to match the solution quality of our methods. Additionally we investigate the support size of the mixed strategies computed by (CG) without operationalizable constraints. Figure 2(b) shows that number of strategies used grows as we increase the problem size (here, the number of flight types). We also compared to a second variation of the column generation method where we pick the top k pure strategies, and compute the optimal mixed strategy over these k strategies. This was done cutting column generation off after 10, 20, 50, 100 columns as well as after full convergence. The results are shown in Figure 3. We see on average a 30% loss in PoU when using this baseline compared to our methods, and in the worst case up to 100% loss with $PoU \sim 2$ for the baseline when compared to our methods. This demonstrates that we can significantly reduce the support size and still obtain a $PoU \sim 1$

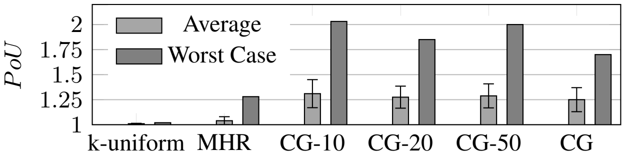

Figure 3: Average case price of usability and b) worst case price of usability, for our two methods (k-uniform and MHR) compared to a cutoff column generation baseline. Column generation (CG) was cutoff after 10, 20 and 50 columns and after convergence.

In Table 1, we compare utility of our algorithms with the utility obtained from solving the full MILP (which optimally solves $\mathscr{P}$k). The full MILP can only be solved for small instances (maximum of $k = 3$). For these instances, we see that both our methods produce near-optimal solutions and can be executed significantly faster. For moderate and large sized instances, we see the k-uniform algorithm outperform MHR in terms of utility, but that MHR can solve large instances faster.

Scalability To evaluate the scalability of our algorithms, we compare the running time for different time windows W and number of passenger categories C. Figure 4 shows the run-

[2] LAX (Los Angeles airport) has an average of 20 unique flight types per terminal (185 destination locations spread over 9 terminals).

	Small		Moderate		Large	
	u*	rt(s)	u*	rt(s)	u*	rt(s)
k-uniform	-85.3	0.2	-543	48	-1258.8	219.4
MHR	-87.0	0.1	-661	20.1	-1315.8	91.2
MILP	-85.2	1154.3	-	-	-	-

Table 1: Runtime and utility u^* of the k-uniform and MHR algorithm compared with the solution of the full MIP (small: $k = 3$, moderate,large: $k = 4$).

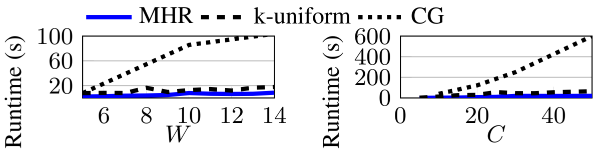

Figure 4: Runtime for different values of W and C ($k = 2$, $\tau = 5$, left: $C = 10$, right: $W = 5$).

ning time for different values of W and C where the rest of the parameters are fixed. This figure shows that the running time is only slightly increasing in W and that our algorithms can be scaled up to a very large number of passenger types.

6 Conclusion

We introduce the new problem of operationalizable strategies in security games and provide a single framework which reasons about the three levels of planning: strategic, tactical and operational level decision problems. Motivated by the important problem of screening for threat we provide algorithmic solutions to overcome the computational challenges that arise when these planning problems are addressed for TSGs and which mitigate the Price of Usability.

References

[AAA, 2014] Transportation security policy. Technical report, American Association of Airport Executives, 2014.

[Balcan *et al.*, 2015] Maria-Florina Balcan, Avrim Blum, Nika Haghtalab, and Ariel Procaccia. Commitment without regrets: Online learning in stackelberg security games. In *Conference on Economics and Computation*. ACM, 2015.

[Basilico *et al.*, 2009] Nicola Basilico, Nicola Gatti, and Francesco Amigoni. Leader-follower strategies for robotic patrolling in environments with arbitrary topologies. In *Autonomous Agents and Multiagent Systems (AAMAS)*, 2009.

[Brown *et al.*, 2016] Matthew Brown, Arunesh Sinha, Aaron Schlenker, and Milind Tambe. One size does not fit all: A game-theoretic approach for dynamically and effectively screening for threats. In *AAAI*, 2016.

[Cooper and Sweller, 1987] G. Cooper and J Sweller. Effects of schema acquisition and rule automation on mathematical problem-solving transfer. *Journal of Educational Psychology*, pages 347–362, 1987.

[Fang *et al.*, 2013] Fei Fang, Albert Xin Jiang, and Milind Tambe. Protecting moving targets with multiple mobile resources. *Journal of Artificial Intelligence Research*, 2013.

[Hogg, 2007] Nanette M. Hogg. Measuring cognitive load. In *Handbook of Research on Electronic Surveys and Measurements*, pages 188–194. Idea Group Inc, 2007.

[Jain *et al.*, 2012] Manish Jain, Kevin Leyton-Brown, and Milind Tambe. The deployment-to-saturation ratio in security games. In *AAAI*, 2012.

[Korzhyk *et al.*, 2010] D. Korzhyk, V. Conitzer, and R. Parr. Complexity of computing optimal Stackelberg strategies in security resource allocation games. In *Proceedings of the 24th AAAI conference on Artificial Intelligence (AAAI)*, pages 805–810, 2010.

[Lipton *et al.*, 2003] Richard J Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *Proceedings of the 4th ACM conference on Electronic commerce*, pages 36–41. ACM, 2003.

[Mc Carthy *et al.*, 2016] Sara Mc Carthy, Milind Tambe, Christopher Kiekintveld, Meredith L. Gore, and Alex Killion. Preventing illegal logging: Simultaneous optimization of resource teams and tactics for security. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, pages 3880–3886. AAAI Press, 2016.

[McCarthy *et al.*, 2017] Sara Marie McCarthy, Phebe Vayanos, and Milind Tambe. Staying ahead of the game: Adaptive robust optimization for dynamic allocation of threat screening resources. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 2017.

[Paas, 1992] F.G. Paas. Training strategies for attaining transfer of problem-solving skill in statistics: A cognitive-load approach. *Journal of Educational Psychology*, pages 429–434, 1992.

[Paruchuri *et al.*, 2007] Praveen Paruchuri, Jonathan P Pearce, Milind Tambe, Fernando Ordonez, and Sarit Kraus. An efficient heuristic approach for security against multiple adversaries. In *Proceedings of the 6th conference on Autonomous Agents and Multiagent Systems*. ACM, 2007.

[Schlenker *et al.*,] Aaron Schlenker, Haifeng Xu, Mina Guirguis, Christopher Kiekintveld, Arunesh Sinha, Milind Tambe, Solomon Sonya, Darryl Balderas, and Noah Dunstatter. Don't bury your head in warnings: A game-theoretic approach for intelligent allocation of cyber-security alerts. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*.

[Schlenker *et al.*, 2016] Aaron Schlenker, Matthew Brown, Arunesh Sinha, and Milind Tambe. Get me to my gate on time: Efficiently solving general-sum bayesian threat screening games. In *ECAI*, 2016.

[Tambe, 2011] Milind Tambe. *Security and game theory: algorithms, deployed systems, lessons learned*. Cambridge University Press, 2011.

[Yin *et al.*, 2015] Yue Yin, Haifeng Xu, Jiarui Gain, Bo An, and Albert Xin Jiang. Computing optimal mixed strategies for security games with dynamic payoffs. In *Proceedings of the 24th International Conference on Artificial Intelligence*, pages 681–687. AAAI Press, 2015.

Proof of Theorem 1

Proof. Given an arbitrary instance of the set cover problem, a number k, a universe $\mathcal{U}$ and a collection of sets $S = \{S_1, S_2, \ldots, S_m\}$ we want to determine if there is a set covering of size k or less ie. a subset $C \subseteq S$, such that $|C| \leq s$. Let G be a game with a number of uniform resources equal to $\max_i |S_i|$, with target set $\mathcal{U}$ and scheduling constraints such that $Q = S$. Let each target have value V, so that the defender receives a utility of $-V$ if the target is unprotected and a utility of zero if the target is protected. Let x_i be the probability that target i is covered by a resource. The defender's expected utility is then $E_x[U] = -\max\limits_{i \in \mathcal{U}} V(1-x_i)$. If there exists an k-operationalizable solution to G with expected utility $E_x[U] > -V$ that means that it is possible for the defender to cover all the targets with some probability using only k pure strategies. These pure strategies in support of the k-operationalizable strategy then give a set cover if size k (or less). Therefore if the corresponding game has an optimal objective value greater than $-V$ there exists a set cover $C \subset S$ such that $|C| \leq k$. □

Proof of Theorem 4

Proof. We give a reduction from the knapsack problem with unbounded items, which is NP-hard. The knapsack problem with unbounded items is described as follows. Given a set of n types of items, where v_j is the value of each item and w_j is the weight of each item, find an allocation of items to the knapsack such that the minimum value over all knapsacks is maximized.

Given an instance of the multiple-knapsack problem with n types value v_i and weights w_j. This can be reduced to the following instance of $\mathscr{P}\mathrm{k}^{LP_n}$. Let $n+1$ be the number of resources. The number of time windows, passenger types and attack methods equal 1, so we omit the indices w, c and m. Also, we choose $k = 1$.

Construct for each resource a team consisting of that resource. The efficiency of each team, E_t, corresponding to resource t equals the value v_t, $t = 1, ..., n$. Additionally, we have one team with resource $n+1$ and the efficiency of this team E_{n+1} equals 0. The capacity of each team, C_r, equals 1.

The number of people required to use one resource, A_r, is given by v_r, $r = 1, ..., n$ and A_{n+1} is chosen to be 0. The maximum number of resources, M_r, is chosen such that $M_r \geq \frac{P}{A_r}$.

We choose the number of passengers arriving, N, such that it can never occur that all passengers can be send to resources $r = 1, ..., n$. The remaining passengers can be send to resource $n+1$. Different choices of N are possible, we choose $N = \max_r M_r$. The utility of a successful attack, U^+, is chosen as N and the utility of an unsuccessful attack, U^-, is 0.

Finally, we choose the value of ϕ_r so high that the overflow variables are always chosen as 0. Note that this model always gives a feasible solution since all remaining passengers be be send to the team with resource $n+1$

Solving $\mathscr{P}\mathrm{k}^{LP_n}$ with the parameters described above gives an optimal solution for the knapsack problem with n types of items, where the resources scheduled corresponds to the items assigned to the knapsack. The optimal value of $\mathscr{P}\mathrm{k}^{LP_n}$ equals the optimal value of the knapsack problem. This can be seen as follows. If a specific resource (item) r, $r = 1, ..., n$ is scheduled, then 1 out of N passengers will be send to that team. So, this resource adds $\frac{E_r}{N}$ $(= \frac{v_r}{N})$ to the total value of z. Since z is multiplied by U^+ $(= N)$, this resource add v_r to the total value of that time window. □

Proof of Theorem 3

Proof. Let G be a zero sum matrix game with payoff matrix R. Let x^*, z^* be a Stackelberg equilibrium solution to game G with game value $(x^*)^\top R z^*$ for the leader. Construct an s-finite adaptive solution x' by sampling from x^*, s times. We want to show that there exists an x' with uniformly distributed support over s strategies and adversary best response y' such that, (x', y') form an ϵ-equilibrium:

$$(x')^\top R z_i - (x')^\top R z' < \epsilon \qquad \forall i \in Q_a, \;\; |\mathcal{Q}_a| = n \tag{5}$$

And the corresponding game value is ϵ away from the Stackelberg equilibrium game value:

$$(x^*)^\top R z^* \geq (x')^\top R z' + \epsilon \tag{6}$$

Let x' be the defender mixed strategy. We construct x' from x^* by drawing s iid samples from the defender pure strategy space according to distribution x^*, and assigning each sample probability $\frac{1}{s}$ in x'. We define the events:

$$E_1 = \{(x^*)^\top R z^* \geq (x')^\top R z' + \epsilon\}$$
$$E_{2,i} = \{(x')^\top R z_i - (x')^\top R z' < \epsilon\}$$

If we can show that the probability of all the events $E_1 \cap_i^n E_{2,i}$ occurring is non zero then there must exist an (x', y') such that these equations hold. What we will do is show that

$$P(E_1 \cap_i^n E_{2,i}) = 1 - P(\neg(E_1 \cap_i^n E_{2,i})) = 1 - P(\neg E_1) - \sum_i^n P(\neg E_{2,i}) > 0$$

Following [Lipton *et al.,* 2003] we define several auxiliary events. Let

$$E_{1,a} = \{(x')^\top R z^* - (x^*)^\top R z^* \geq \frac{\epsilon}{2}\} \tag{7}$$

$$E_{1,b} = \{(x')^\top R z' - (x^*)^\top R z' \geq \frac{\epsilon}{2}\} \tag{8}$$

We note (as in [Lipton *et al.,* 2003]) that

$$\begin{aligned} E_1 &\supseteq \;\; E_{1,a} \cap E_{1,b} \\ P(E_1) &\geq \;\; P\left(E_{1,a} \wedge E_{1,b}\right) \\ P(\neg E_1) &\leq \;\; P(\neg_{1,a}) + P(\neg E_{1,b}) \end{aligned} \tag{9}$$

Similarly define:

$$E_{2,a} = \{(x')^\top R z' - (x^*)^\top R z' < \frac{\epsilon}{2}\}$$

$$E_{2,i}^b = \{(x')^\top R z^i - (x^*)^\top R z^i < \frac{\epsilon}{2}\}$$

so that:

$$\begin{aligned} E_{2,i} &\supseteq \;\; E_{2,a} \cap E_{2,i}^b \\ P(E_{2,i}) &\geq \;\; P(E_{2,a} \wedge E_{2,i}^b) \\ P(\neg E_{2,i}) &\leq \;\; P(\neg E_{2,a}) + P(\neg E_{2,i}^b) \end{aligned} \tag{10}$$

Again following [Lipton *et al.,* 2003] we note that $(x')^\top R z$ is a sum of k independent random variables each of expected value $(x^*)^\top R z$ with each random variable having value between 0 and 1. We can apply a standard concentration bounds (using Hoeffding's inequality) and get:

$$P(\neg E_{1,a}) \leq e^{-\frac{k\epsilon^2}{2}}$$
$$P(\neg E_{1,a}) \leq e^{-\frac{k\epsilon^2}{2}}$$
$$P(\neg E_1) \leq 2e^{-\frac{k\epsilon^2}{2}}$$

Similarly:

$$P(\neg E_{2,a}) \leq e^{-\frac{k\epsilon^2}{2}}$$
$$P(\neg E_{2,i}^b) \leq e^{-\frac{k\epsilon^2}{2}}$$
$$P(\neg E_{2,i}) \leq 2e^{-\frac{k\epsilon^2}{2}}$$

So that finally:

$$\begin{aligned} P(\neg E_1 \cap_i^n E_{2,i}) &\leq 2e^{-\frac{k\epsilon^2}{2}} + 2ne^{-\frac{k\epsilon^2}{2}} \\ &\leq 2(1+n)e^{-\frac{k\epsilon^2}{2}} \end{aligned}$$

For $k > \frac{4log(1+n)}{\epsilon^2}$ we then have that $P(\neg(E_1 \cap_i^n E_{2,i})) < 1$ and $P(E_1 \cap_i^n E_{2,i}) > 0$ which concludes the proof.

□