Cyber Camouflage Games for Strategic Deception

Omkar Thakoor[1], Milind Tambe[1], Phebe Vayanos[1], Haifeng Xu[2], Christopher Kiekintveld[3], and Fei Feng[4]

[1] University of Southern California, Los Angeles, CA 90007, USA
(othakoor, tambe, phebe.vayanos)@usc.edu
[2] University of Virginia, Charlottesville, VA 22904, USA
hx4ad@virginia.edu
[3] University of Texas at El Paso, El Paso, TX 79968, USA
cdkiekintveld@utep.edu
[4] Carnegie Mellon University, Pittsburgh, PA 15213, USA
feifang@cmu.edu

Abstract. The rapid increase in cybercrime, causing a reported annual economic loss of $600 billion (Lewis, 2018), has prompted a critical need for effective cyber defense. Strategic criminals conduct network reconnaissance prior to executing attacks to avoid detection and establish situational awareness via scanning and fingerprinting tools. Cyber deception attempts to foil these reconnaissance efforts by camouflaging network and system attributes to disguise valuable information. Game-theoretic models can identify decisions about strategically deceiving attackers, subject to domain constraints. For effectively deploying an optimal deceptive strategy, modeling the objectives and the abilities of the attackers, is a key challenge. To address this challenge, we present Cyber Camouflage Games (CCG), a general-sum game model that captures attackers which can be diversely equipped and motivated. We show that computing the optimal defender strategy is NP-hard even in the special case of *unconstrained* CCGs, and present an efficient approximate solution for it. We further provide an MILP formulation accelerated with cut-augmentation for the general *constrained* problem. Finally, we provide experimental evidence that our solution methods are efficient and effective.

Keywords: Game Theory, Cyber Deception, Optimization

1 Introduction

The ubiquity of Internet connectivity has spurred a significant increase in cybercrime. Major cyber attacks such as recent data breaches at Equifax (Gutzmer, 2017), Yahoo (Goel and Perlroth, 2016), as well as government agencies like the Office of Personnel Management (Peterson, 2015) are often executed by adept attackers conducting reconnaissance as the first stage for an effective cyber attack (Mandiant, 2013; Joyce, 2016). Rather than attempting "brute force" exploits, scanning tools such as NMap (Lyon, 2009), xProbe2 (Arkin and Yarochkin, 2003) and fingerprinting techniques such as sinFP (Auffret, 2010) are used to identify vulnerabilities and develop specific plans to infiltrate the network while minimizing the risk of detection.

To mitigate the reconnaissance abilities of attackers, *deception* techniques aim to disguise valuable network information to create uncertainty. This can lead attackers to

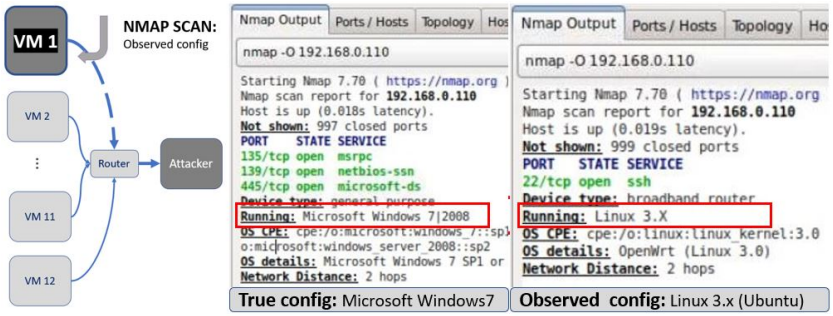


Fig. 1: The attacker scans virtual machines on the test-bed and the configuration observed via NMAP can be set to differ from the true configuration that NMAP would show without our deployed deception.

spend more time in reconnaissance activities (increasing the chances of detection), or to attempt infiltration tactics that are less effective. Examples of such techniques include the use of honeypots or decoys (Ferguson-Walter et al, 2017), real systems using deceptive defenses (De Gaspari et al, 2016), and obfuscated responses to fingerprinting (Berrueta, 2003; Rahman et al, 2013). Canary (Thinkst, 2015) is an example of a deception-based tool in commercial use, while CyberVAN (Chadha et al, 2016) is a test-bed for simulating various deception algorithms. Fig. 1 shows a demonstration of our model on CyberVAN.

There are two general factors to consider when deploying cyber deception techniques. First, strategic use of deception is vital due to the significant costs and feasibility constraints that must be considered; e.g., deception via counter-fingerprinting techniques like HoneyD, OSfuscate, and IPMorph typically degrades performance (Rahman et al, 2013). In most deception methods, we must also consider the costs of deploying, and maintaining deceptive strategies which may include both computational resources and developer time.

Second, optimizing the effectiveness of deception depends on modeling the preferences and capabilities of the attacker. The attacker's goals can greatly vary — they may exactly conflict the defender's, or they could be partially orthogonal. For instance, an economically motivated attacker may find utility primarily in financial records whereas the defender may consider losing national security data as a more critical loss. In many cases, the preferences may be strongly governed by the available exploits. Despite this diversity in real-world adversaries, previous game theoretic models for cyber deception assume zero-sum payoffs, implying directly conflicting attacker motives. Hence, to eliminate this fundamental limitation, this paper considers general-sum payoffs in this setting. Furthermore, for situations where there may be uncertainty in the defender's knowledge of attacker's payoffs, this work serves as a vital stepping stone to model such uncertainty (more details on related work in Sec. 1.1).

The main contributions of this paper are as follows. First, we present Cyber Camouflage Games — a general-sum game model that presents completely distinct computational challenges and insights relative to the previous zero-sum models. Second,

we prove that computing an optimal solution is NP-hard even for *unconstrained* CCG and present a Fully Polynomial Time Approximation Scheme (FPTAS) for this case. Third, for CCG with constraints, we present an MILP formulation to find an optimal solution, harnessing *polytopal* strategy space for compactness and boosted with cut augmentation. Finally, we experimentally evaluate our algorithms and show substantial improvement in scalability and robustness.

1.1 Related work

The Cyber Deception Game (CDG) (Schlenker et al, 2018) is a game-theoretic deception model limited to *zero-sum* settings. It cannot model diversely modeled attackers and only focuses on the challenge of deception being costly and partly infeasible but fails to present the strategic challenge which exists regardless of top-end deception methods, which our model highlights. Since a zero-sum model also implicitly implies perfectly known payoffs which may not always be possible, eliminating this fundamental limitation by considering general-sum payoffs as we do, can allow for a more holistic model in the future that considers uncertainty when making decisions, as many security game models previously have (Kiekintveld et al, 2011, 2013; Nguyen et al, 2014).

Other works in cyber defense (Alpcan and Başar, 2010; Laszka et al, 2015; Serra et al, 2015; Schlenker et al, 2017) have adopted game theoretic models, including several that aim to strategically deploy honeypots (Pıbil et al, 2012; Durkota et al, 2015). However, these do not consider camouflaging the network as in our model. De Gaspari et al (2016) provide a realistic systems architecture for active defense using the same types of deception abilities we consider, but they do not address how to strategically optimize these tactics under practical constraints. Several use moving target defense that mitigate attacker reconnaissance by using movement to adapt and randomize the attack surface of a network or system (Albanese et al, 2014; MacFarland and Shue, 2015; Albanese et al, 2016; Achleitner et al, 2016), but this work typically does not model nor optimize against a strategic adversary.

Despite being a Stackelberg game for a security domain, CCGs have a very distinct structure in comparison with Stackelberg Security Games (SSG) (Tambe, 2011), since the core defensive action of "masking" differs from "defending" targets in SSG in several ways. First, security resources are limited in SSGs, while in CCG every target can be masked. Second, covering a target in SSGs directly improves its security, whereas in CCGs, the effectiveness of masking depends on how other machines are masked to alter the attacker's information state. Finally, SSGs typically focus on mixed strategies and the Strong Stackelberg Equilibria (SSE), whereas CCGs are restricted to pure strategies and therefore need the Weak Stackelberg Equilibrium (WSE) concept. Pita et al (2010) present a robust approach for sub-optimal attackers that can be adapted for WSE computation in normal-form Stackelberg games, but cannot be directly applied to CCGs due to the exponential strategy space.

2 Cyber Camouflage Games

We refer to a network administrator as the "defender" and a cybercriminal as the "attacker". CCGs have the components explained as follows.

Network Configurations. A network consists of a set of machines indexed in $\mathcal{K} := \{1, \ldots, |\mathcal{K}|\}$. Each machine has a *true configuration* (TC) which is modeled as a tuple of attributes such as [OS Linux, Webserver TomCat 8]. The TC should be a complete description of the security relevant features of the machine, so machines with the same TC are considered identical. Let $\mathcal{I}$ index the set of TCs present in the network. The *true state of the network* (TSN) is defined by a vector $\boldsymbol{n} = (n_i)_{i \in \mathcal{I}}$ where each n_i denotes the number of machines with TC i.

Using deception techniques, the defender can disguise each machine by obfuscating some of its attributes. We say the defender "masks" each machine with an *observed configuration* (OC); $\mathcal{J}$ denotes the set of all possible OCs. An OC similarly captures the set of observed attributes, e.g., [OS Linux, Webserver Nginx 1.8], and is assumed to be a complete representation of the information observed by the attacker so machines with the same OC are indistinguishable to the attacker. This framework can directly capture deception via obfuscation of system attributes, and is also applicable to other deception methods such as honeypots by including a "honeypot" as a TC, and the configurations it mimics as OCs.

Deception Strategies. The defender's strategy can be encoded as an integer matrix Φ, where Φ_{ij} denotes the number of machines with TC i, masked with OC j. The *observed state of the network* (OSN) is a vector that, unlike the TSN, is a function of the strategy Φ. We denote an OSN as $\boldsymbol{m}(\Phi) := (m_j(\Phi))_{j \in \mathcal{J}}$, where $m_j(\Phi) = \sum_i \Phi_{ij}$ denotes the number of machines masked by OC j for strategy Φ.

Strategy feasibility and costs. Achieving deception is often costly and not arbitrarily feasible. We represent *feasibility* constraints using a (0,1)-matrix Π, where $\Pi_{ij} = 1$ if TC i can be masked with OC j. Further, $\mathcal{J}_i$ denotes the set of OCs that can mask TC i, i.e., $\mathcal{J}_i := \{j \in \mathcal{J} \mid \Pi_{ij} = 1\}$. Next, we assume that masking a TC i with an OC j, has a cost of c_{ij} incurred by the defender — this is relevant only if $\Pi_{ij} = 1$, and denotes the combined costs from deployment, maintenance, degraded functionality, etc. The defender can afford the total cost of masking up to a *budget* B. Let $\mathcal{F}$ denote the set of strategies that are *feasible* and *affordable* — described with linear constraints:

$$\mathcal{F} = \left\{ \Phi \;\middle|\; \begin{array}{l} \Phi_{ij} \in \mathbb{Z}_{\geq 0}, \quad \Phi_{ij} \leq \Pi_{ij} n_i \; \forall (i,j) \in \mathcal{I} \times \mathcal{J}, \\ \sum\limits_{j \in \mathcal{J}} \Phi_{ij} = n_i \;\; \forall i \in \mathcal{I}, \quad \sum\limits_{i \in \mathcal{I}} \sum\limits_{j \in \mathcal{J}} \Phi_{ij} \; c_{ij} \leq B \end{array} \right\}$$

The first and the third constraints follow from the definition of Φ and $\boldsymbol{n}$ resp. The second inequality imposes the feasibility constraints, and the fourth, the budget constraint.

Defender and Attacker Valuations. If a machine with TC i is attacked, the attacker gets a utility v_i^{a} — his *valuation* of TC i. Collectively, these are represented as a vector $\boldsymbol{v}^{\mathbf{a}}$. Analogously, we define valuations $\boldsymbol{v}^{\mathbf{d}}$ for the defender; a higher v_i^{d} reflects a smaller loss when TC i is compromised.

Remark 1. It is natural to set v^{a} as positive values and v^{d} as negative ones, however, the problem remains equivalent (as explained momentarily) if the defender valuations (or, independently, the attacker valuations as well) are simultaneously scaled by a positive constant or shifted by a constant, so we do not specify positivity of any values.

Game Model. We model the interaction as a Stackelberg game to capture the sequence of decisions between the players. The defender is the leader who knows the TSN

$\boldsymbol{n}$ and can deploy a deception strategy Φ. The attacker observes the OSN and chooses an OC to attack. Since the attacker cannot distinguish machines with the same OC, this is interpreted as an attack on a randomly selected machine with this OC.

We assume that the defender can only play a pure strategy since it is usually not possible to change the network frequently, making the attacker's view of the network static. We assume the attacker perfectly knows the defender's strategy Φ to compute best response, as in CDG (Schlenker et al, 2018), which is justified via insider information leakage or other means of surveillance.

When the defender plays a strategy Φ, her expected utility when OC j is attacked (with $m_j(\Phi) > 0$), is given by

$$u^{\mathrm{d}}(\Phi, j) = \mathbb{E}[v_i^{\mathrm{d}} | \Phi, j] = \sum_{i \in \mathcal{I}_j} \mathbb{P}(i|\Phi, j) v_i^{\mathrm{d}} = \sum_{i \in \mathcal{I}} \frac{\Phi_{ij}}{m_j(\Phi)} v_i^{\mathrm{d}}.$$

Here, $\mathbb{E}[\cdot]$ denotes the expectation operator, and $\mathbb{P}$, the probability of TC of the attacked machine, conditioned on its OC j and the defender strategy Φ. Similarly, the attacker's expected utility in this case is $u^{\mathrm{a}}(\Phi, j) = \sum_{i \in \mathcal{I}} \frac{\Phi_{ij}}{m_j(\Phi)} v_i^{\mathrm{a}}$. These utility expressions justify Remark 1.

An illustrative example of CCGs is as follows.

CCG Example: Consider a CCG with 6 machines, 4 TCs and 3 OCs. Let the TSN be $\boldsymbol{n} = (2, 2, 1, 1)$. Let the valuations be $\boldsymbol{v}^{\mathbf{d}} = (8, 2, 7, 11)$ and $\boldsymbol{v}^{\mathbf{a}} = (7, 2, 5, 11)$. Let $\mathcal{J}_1 = \{1\}$, $\mathcal{J}_2 = \{2\}$, $\mathcal{J}_3 = \{1, 3\}$ and $\mathcal{J}_4 = \{2, 3\}$. Let the costs be $c_{31} = 5$, and $c_{ij} = 1$ for all other feasible (i, j) pairs, and let the budget $B = 7$. Thus, machines with TC 1 and 2 have only 1 choice of OC to mask due to feasibility constraint. Masking TC 3 with OC 1 at cost 5 is too expensive, since masking the remaining machines costs at least 3. Thus, due to the budget constraint, TC 3 has OC 3 as the unique choice. Thus, the defender's strategy space is

$$\mathcal{F} = \left\{ \Phi = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \Phi' = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \right\}$$

If the defender plays Φ, attacker's best response is to attack OC 1, yielding expected utilities $u^{\mathrm{a}}(\Phi, 1) = 7$, and $u^{\mathrm{d}}(\Phi, 1) = 8$ for the attacker and the defender, resp.

Optimization problem Having defined the game model, we now discuss the solution approach. Previous work on general-sum Stackelberg games has typically used *Strong Stackelberg equilibria* (SSE). This assumes that whenever the follower has multiple best responses, he breaks ties in favor of the leader (i.e., maximizing her payoff), which the defender can *induce* using mixed strategies. The defender cannot always induce a specific response in a CCG since he is restricted to pure strategies (Guo et al, 2018).

Therefore, we consider the *robust* assumption that the attacker breaks ties against the defender. This worst-case tiebreaking leads to *Weak Stackelberg Equilibria* (WSE) (Breton et al, 1988). A drawback of WSE is that it may not exist (von Stengel and Zamir, 2004). However, it has been shown to exist when the defender can play a finite set of pure strategies as in CCG. We therefore adopt WSE and assume that the attacker

chooses a best response to the defender strategy Φ, minimizing the defender utility in case of a tie. Thus, the defender's utility is $u^{\min}(\Phi)$ as defined by the following optimization problem (OP):

$$\min_{j} \; u^{\mathrm{d}}(\Phi, j) \mid u^{\mathrm{a}}(\Phi, j) \geq u^{\mathrm{a}}(\Phi, j') \;\; \forall j' \in \mathcal{J}. \tag{1}$$

Hence, the defender needs to choose $\mathrm{argmax}_{\Phi}\, u^{\min}(\Phi)$.

We first study the game without any feasibility or budget constraints. This *unconstrained* CCG underlines the inherent challenge of strategic deception even when sophisticated techniques are available for arbitrarily masking TCs with any OCs at low costs.

Remark 2. Note that, setting all entries of Π to 1 makes all strategies feasible, and setting the budget and costs so that $B \geq \sum_i n_i \max_{j \in \mathcal{J}_i} c_{ij}$ makes all feasible strategies affordable.

3 Unconstrained CCGs

In this setting, masking any TC with any OC is possible, and every feasible strategy has total cost within budget. First, we prove that

Theorem 1. *Computing the optimal defender strategy in unconstrained CCGs is NP-hard.*

Proof. We prove the NP-hardness via a reduction from the subset sum problem, denoted as `SubsetSum`, which is a well-known NP-complete problem. Given a set S of *integers*, `SubsetSum` is the decision problem to determine whether there is a non-empty *subset* of S whose sum is zero.

An instance of `SubsetSum` is specified by a set of N integers $\{x_1, \ldots, x_N\} = S$ (w.l.o.g., assume $x_n \neq 0$ for any n and $\sum_{n\in[N]} x_n \neq 0$ since otherwise the problem is trivial). Given such an instance of `SubsetSum`, we construct the following unconstrained general-sum CDG. First, let $w = -\sum_{n\in[N]} x_n \in \mathbb{Z}$, so that, $\sum_i x_i + w = 0$. Let $\delta_1, \delta_2 \in (0, 1)$ be small constants s.t. $1 - \delta_2 \geq \delta_1 \geq (N+1)\delta_2$. We construct a CDG with $(N+2)$ machines each with a different TC. Thus, $\mathcal{K} = \mathcal{I} = \{1, \ldots, N, N+1, N+2\}$, and the TSN $\boldsymbol{n}$ is the all-one vector. There are two OCs, i.e., $\mathcal{J} = \{1, 2\}$. The valuations for the attacker and the defender are defined as $\boldsymbol{v}^{\mathbf{a}} = (x_1, \ldots, x_N, w, \delta_2)$ and $\boldsymbol{v}^{\mathbf{d}} = (-x_1, \ldots, -x_N, -w + \delta_1, \delta_2)$, respectively. We remark that, the defender's and attacker's valuations are only non-zero-sum on the last two TCs. This completely defines an unconstrained CDG instance.

We claim that the defender can achieve utility strictly greater than $\frac{\delta_1+\delta_2}{N+2}$ in the constructed instance if and only if the `SubsetSum` instance is a YES instance. As a result, any algorithm for computing the optimal defender utility for general-sum CDGs can be transferred, in polynomial time, to an algorithm for `SubsetSum`. This implies the NP-hardness of solving unconstrained CDGs.

We first show that if the `SubsetSum` is a YES instance, then the defender can achieve a utility strictly greater than $\frac{\delta_1+\delta_2}{N+2}$. By assumption, there exists a non-empty

set $S' \subset S$ such that $\sum_{x_n \in S'} x_n = 0$. Let $N' = |S'| > 0$. Consider the strategy that masks all TCs in $\mathcal{I}' = \{i \mid x_i \in S'\}$ to OC 1, and masks TCs in $\mathcal{I} \setminus \mathcal{I}'$ to OC 2. By construction, the attacker will have expected utility 0 on OC 1 but a strictly positive utility on OC 2. As a result, the attacker will attack OC 2, resulting in expected defender utility $\frac{\delta_1+\delta_2}{N+2-N'} > \frac{\delta_1+\delta_2}{N+2}$. As a result, the optimal defender utility must be strictly greater than $\frac{\delta_1+\delta_2}{N+2}$.

Next, we show that if the `SubsetSum` is a NO instance, then the optimal defender utility is at most $\frac{\delta_1+\delta_2}{N+2}$. We consider the following particular masking strategies:

1. If all the TCs are masked by one OC, then the defender will achieve expected utility $\frac{-\sum_n x_n - w + \delta_1 + \delta_2}{N+2} = \frac{\delta_1+\delta_2}{N+2}$
2. If machine $N+2$ is masked by (say) OC 1 and all other machines are masked by OC 2, then the attacker has a better utlity in attacking OC 1, resulting in defender utility $\delta_2 \leq \frac{\delta_1+\delta_2}{N+2}$ by construction.
3. Otherwise, any other solution to the CDG instance corresponds to a partition of $\mathcal{I}$ into two non-empty sets, denoted as $\mathcal{I}_1, \mathcal{I}_2$, which correspond to TCs masked as OC $1, 2$ respectively. Moreover, w.l.o.g., assume $N+2 \in \mathcal{I}_1$ and $|\mathcal{I}_1| > 1$. Then $\mathcal{I}_2$ is a strict subset of $\mathcal{I} \setminus \{N+2\}$ which all have integer attacker values. Since the `SubsetSum` is a NO instance, we know that the total attacker values in $\mathcal{I}_2$ cannot sum up to 0. If they sum up to a positive integer (thus at least 1), then two properties hold: 1. The total *attacker* value in $\mathcal{I}_1$ is at most $-1 + \delta_2 < 0$; 2. The total *defender* value in $\mathcal{I}_2$ is at most $-1 + \delta_1 < 0$. The first property implies that the attacker will attack OC 2 and the second property implies that the defender will get strictly negative utility. Similarly, if the total attacker values in $\mathcal{I}_2$ sum up to a negative integer, the attacker will attack OC 1, still resulting in a negative defender utility. To sum up, in this case, the optimal defender utility is at most $\frac{\delta_1+\delta_2}{N+2}$. This concludes the proof.

Thus, this result is in sharp contrast to unconstrained CDGs where masking all the machines with the same OC is an optimal strategy and thus the computation has constant-time complexity.

We now show that despite the NP-hardness, the problem admits a *Fully Polynomial Time Approximation Scheme* (FPTAS). To that end, we first need the following proposition.

Proposition 1. *Unconstrained CCGs always have an optimal defender strategy where at most 2 OCs mask all the machines.*

Proof. Let OC 1, 2 be feasible for all the TCs. Let Φ be any optimal strategy which yields defender utility u. Let $\mathcal{J}' = \operatorname{argmax}_j u^{\mathrm{a}}(\Phi, j)$ denote the set of attacker's best response OCs. Then consider a strategy Φ^* as follows: $\Phi^*_{i1} = \sum_{j' \in \mathcal{J}'} \Phi_{ij'}$, $\Phi^*_{i2} = \sum_{j' \notin \mathcal{J}'} \Phi_{ij'}$ $\forall\ i \in \mathcal{I}$, and $\Phi^*_{ij} = 0$ for all other OC $j \neq 1, 2$. Then, Φ^* induces the attacker to attack OC 1, resulting in defender utility at least u (as the defender utility for every OC in $\mathcal{J}'$ is at least u). Thus, Φ^* is optimal and uses at most 2 OCs to mask all the machines.

Next, assume, w.l.o.g., that $v^{\mathrm{d}}_i \in [0, 1]$ $\forall i$ (as the problem is equivalent if the valuations are shifted, or, simultaneously scaled by a positive constant). Then, we show an FPTAS:

Theorem 2. *For any $\epsilon > 0$, there is a $O(n^3/\epsilon)$ time algorithm that computes a deception strategy with defender utility at most ϵ less than the optimal.*

Proof. We use dynamic programming (DP) to compute an approximate solution. To start, we first discretize the defender valuations by rounding them *down* to the closest multiples of ϵ. Let integer $v_i = \lfloor v_i^{\mathrm{d}}/\epsilon \rfloor$, so that $v_i\epsilon$ is the defender valuation rounded down. Note that $v_i \in [0, 1/\epsilon]$.

By Proposition 4.4, we can w.l.o.g. focus on strategies using the 2 OCs to mask all the machines. We design the strategy such that OC 1 is the attacker's best response. Our idea is to compute a 3-dimensional table A, where $A[i, k, l]$ denotes the *maximum attacker valuation sum* for attacking OC 1, over all the strategies in which OC 1 masks exactly k machines, all from among the first i machines with the *defender valuation sum* being $l\epsilon$ for OC 1. By definition, $A[i, k, l]$ satisfies the following recurrence relation:

$$A[i, k, l] = \max\{A[i-1, k-1, l-v_i] + v_i^{\mathrm{a}}, A[i-1, k, l]\}$$

which follows from considering the two options for machine i — whether to mask it with OC 1 or not. The base cases are $A[0, 0, 0] = 0$ and $A[0, k, l] = -\infty$ if either k or l are non-zero.

After computing table A, we are ready to compute the optimal defender strategy w.r.t. the rounded defender payoffs. In particular, the maximum defender utility of our strategy is the maximum value of $l\epsilon$ such that $\exists k > 0$ with attacker's utility for attacking OC 1, i.e., $A[n, k, l]/\,k$, being more than that of attacking OC 2, or equivalently, more than the average attacker valuation $\sum_i v_i^{\mathrm{a}} n_i/|\mathbf{n}|$. Such a table entry can be found by enumerating $A[n, k, l]$ for different k and l.

It is easy to see that the DP computes an optimal defender strategy for defender payoff $\boldsymbol{v}$. To prove that this strategy is an additive ϵ approximation to the original problem with defender payoffs $\boldsymbol{v}^{\mathbf{d}}$, let $U^d(\boldsymbol{u}, \Phi)$ denote the defender utility when using defender valuations $\boldsymbol{u}$ and strategy Φ. Let Φ^* denote the optimal strategy to the original problem and $\hat{\Phi}$ be the strategy output by our algorithm. We have

$$U^d(\boldsymbol{v}^{\mathbf{d}}, \hat{\Phi}) \geq U^d(\boldsymbol{v}, \hat{\Phi}) \geq U^d(\boldsymbol{v}, \Phi^*) \geq U^d(\boldsymbol{v}^{\mathbf{d}}, \Phi^*) - \epsilon$$

where the first and third inequalities are due to the rounding down of $\boldsymbol{v}^{\mathbf{d}}$ to $\boldsymbol{v}$ entry-wise by ϵ, and the second inequality follows by optimality of $\hat{\Phi}$. This concludes the proof.

4 Constrained CCGs

4.1 Optimal defender strategy MILP formulation.

Our goal is to compute the WSE, i.e., to compute $\max_{\Phi \in \mathcal{F}} u^{\min}(\Phi)$. As $u^{\min}(\Phi)$ is given by OP (1), computing WSE is a bilevel OP which cannot ordinarily be reduced to a single-level Mixed Integer Linear Program (MILP) (Sinha et al, 2018). In particular, the single-level reduction has been shown for SSE computation, since the attacker's tiebreaking aligns with the defender's objective. However, this does not apply to WSE

due to the worst-case tiebreaking. Hence, we first formulate an OP which considers ϵ-optimal responses for the attacker (for a small constant ϵ) and assume he selects the one with the least defender utility. This OP (referred to as GS-MIP) is:

$$\max_{\Phi,\boldsymbol{q},\gamma,\alpha} \ \gamma \tag{2}$$

$$\text{s.t.} \quad \alpha, \gamma \in \mathbb{R},\ \Phi \in \mathcal{F},\ \boldsymbol{q} \in \{0,1\}^{|\mathcal{J}|}$$

$$q_1 + \ldots + q_{|\mathcal{J}|} \geq 1 \tag{2a}$$

$$\epsilon(1 - q_j) \leq \alpha - u^{\mathrm{a}}(\Phi, j) \qquad \forall j \in \mathcal{J} \tag{2b}$$

$$M(1 - q_j) \geq \alpha - u^{\mathrm{a}}(\Phi, j) \qquad \forall j \in \mathcal{J} \tag{2c}$$

$$\gamma \leq u^{\mathrm{d}}(\Phi, j) + M(1 - q_j) \qquad \forall j \in \mathcal{J} \tag{2d}$$

$$q_j \leq m_j(\Phi) \qquad \forall j \in \mathcal{J}. \tag{2e}$$

The maximization objective γ gives the defender's optimal utility. The binary variables q_j indicate if attacking OC j is an ϵ-optimal attacker strategy, of which there is at least one an possibly more, as specified by (2a). (2b) and (2c) make α the optimal attacker utility, and enforce $q_j = 1$ for all the ϵ-optimal strategies for the attacker (using a big-M constant). (2e) ensures that only the OCs which actually mask a machine are considered as valid attacker responses. Finally, (2d) captures the worst-case tiebreaking by requiring that γ is least of the utilities the defender can get from a possible ϵ-optimal attacker response. [5]

In reality, an optimal attacker corresponds to having $\epsilon = 0$ by definition. Nevertheless, setting $\epsilon > 0$ is necessary to enforce the worst-case tiebreaking as explained above for constraint (2b); setting $\epsilon = 0$ can be shown to lead to an SSE solution, and not WSE. Despite this challenge, since the number of targets are finite, there must be an ϵ such that only the optimal strategies are ϵ-optimal. Then, for such small enough ϵ, (2b) would enforce that the attacker can choose from precisely the set of optimal strategies. Hence, we conclude that,

Proposition 2. *$\exists \epsilon > 0$ s.t. OP (2) computes $\max_{\Phi \in \mathcal{F}} u^{\min}(\Phi)$.*

Remark 3. ϵ should be set to a value that ensures that the second-best attacker utility is at least epsilon less than optimal. It suffices to set it to L/k^2 where L is the bit precision of the valuations and $k = |\mathcal{K}|$ is the number of machines.

Other works considering ϵ-optimal responses include (Tijs, 1981) which computes ϵ-optimal Nash equilibria and (Pita et al, 2010) which considers robust optimization against boundedly rational opponents in Bayesian normal-form Stackelberg games. Despite similarities, in particular, that of a Stackelberg setting in the latter, their solution methods do not apply here due to key differences in the CCG model, viz., non-Bayesian setting, perfect rationality, restriction to pure strategies, and most importantly, compact input representation via *polytopal* strategy space (Jiang et al, 2017). This makes it non-viable to enumerate strategies like normal-form games. However, the utility functions

[5] The additional constant M can be simply replaced by $\max_{i,i'} |v_i^{\mathrm{a}} - v_{i'}^{\mathrm{a}}|$ and $\max_{i,i'} |v_i^{\mathrm{d}} - v_{i'}^{\mathrm{d}}|$ resp. in the 3rd, 4th constraints

u^{d} and u^{a} are *linear fractionals*, i.e., ratios of expressions that are linear in Φ. This property allows for an MILP formulation of GS-MIP despite the structural complexity of CCGs, as follows.

We use an alternate representation of the defender's strategy with a $|\mathcal{K}| \times |\mathcal{J}|$ (0,1)-matrix Θ, where $\Theta_{kj} = 1$ iff machine k is masked with OC j. Then, we can write the OSN $\boldsymbol{m}$ as $m_j(\Theta) = \sum_{k \in \mathcal{K}} \Theta_{kj}$, and the player utilities as,

$$u^{\mathrm{d}}(\Theta, j) = \frac{\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \Theta_{kj} v_i^{\mathrm{d}}}{m_j(\Theta)} \; ; u^{\mathrm{a}}(\Theta, j) = \frac{\sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \Theta_{kj} v_i^{\mathrm{a}}}{m_j(\Theta)},$$

where, $\mathcal{K}_i$ is the set of machines with TC i. Substituting these expressions in constraints of GS-MIP, e.g., say (2b), and multiplying the equation to get rid of fractional expressions yields,

$$\epsilon(1 - q_j) \sum_k \Theta_{kj} \leq \alpha \sum_k \Theta_{kj} - \sum_{i \in \mathcal{I}} \sum_{k \in \mathcal{K}_i} \Theta_{kj} v_i^{\mathrm{a}} \quad \forall j \in \mathcal{J}.$$

The constraint above and the ones obtained by similarly transforming (2c), (2d), (2e), contain bilinear terms that are products of a binary variable with a binary or continuous variable, which can be linearized using standard techniques. The complete resultant MILP can be found in the appendix.

4.2 Cuts to Speed up the MILP Formulation

Symmetry breaking. Using the alternate representation Θ exponentially blows up the feasibility region of the MILP, since each strategy Φ has many equivalent Θ representations due to machines having the same TC being identical. For instance, a strategy Φ which masks the n_i machines having TC i with a different OC each, results in $n_i!$ equivalent Θ representations corresponding to the different permutations of machine assignment to OCs. To break this symmetry, we add constraints to require that the assignment of machines to OCs is lexicographically sorted within a TC, i.e., for machines k, k' with the same TC, masked with different OCs $\hat{j}$ and j' resp., we must have $k < k' \Leftrightarrow \hat{j} < j'$. A linear constraint captures this for machines k, k':

$$\sum_{j \in \mathcal{J}} j \Theta_{kj} \leq \sum_{j \in \mathcal{J}} j \Theta_{k'j}. \tag{3}$$

Proposition 3. *For any strategy Θ, and for machines k, k', masked with OCs $\hat{j}$ and j' as per Θ,* (3) $\Leftrightarrow \hat{j} < j'$. *Further, adding constraints* (3) *to GS-MIP preserves at least one optimal solution while eliminating all of the symmetric solutions.*

Proof. For any strategy Θ, for any machine k, we must have $\sum_{j \in \mathcal{J}} \Theta_{kj} = 1$. Suppose machines k, k' are masked with different OCs $\hat{j}$ and j'. Thus, $\Theta_k = \mathbf{e}_{\hat{j}}$, and $\Theta_{k'} = \mathbf{e}_{j'}$

(where $\mathbf{e}_j$ denotes the unit vector with 1 in the j^{th} coordinate and 0 elsewhere). Hence, $\sum_{j\in\mathcal{J}} j\Theta_{kj} = \hat{j}$ and $\sum_{j\in\mathcal{J}} j\Theta_{k'j} = j'$. Thus, it follows that

$$\sum_{j\in\mathcal{J}} j\Theta_{kj} < \sum_{j\in\mathcal{J}} j\Theta_{k'j} \Leftrightarrow \hat{j} < j'$$

To ensure the lexicographically sorted assignment, it suffices to add constraint (3) for only the pairs k, k' which are consecutively indexed within each TC, i.e., $|\mathcal{K}_i| - 1$ constraints for TC i, and thus, linearly many in total.

Bounding attacker's optimal utility. The attacker's optimal utility α, by definition, is at least the utility $u^{\text{a}}(\Theta, j)$ of attacking any OC j (follows from (2b)). Consequently, it must be at least the average utility of all machines, i.e.,

Proposition 4. $\alpha \geq \frac{1}{|\mathcal{K}|}\sum_{i\in\mathcal{I}} n_i v_i^{\text{a}}$.

Proof. By constraint (2b*), we have,

$$\epsilon(1-q_j)\sum_k \Theta_{kj} \leq \alpha\sum_k \Theta_{kj} - \sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}\Theta_{kj}v_i^{\text{a}} \qquad \forall j\in\mathcal{J}$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}\Theta_{kj}v_i^{\text{a}} \leq \alpha\sum_k \Theta_{kj} \qquad \forall j\in\mathcal{J}$$

($\because \epsilon,\ \mathbf{q},\ \Theta$ are non-negative by definition)

$$\sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}\Theta_{kj}v_i^{\text{a}} \leq \sum_{j\in\mathcal{J}}\alpha\sum_k\Theta_{kj}$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i}\sum_{j\in\mathcal{J}}\Theta_{kj}v_i^{\text{a}} \leq \alpha\sum_k\sum_{j\in\mathcal{J}}\Theta_{kj}$$

$$\sum_{i\in\mathcal{I}}\sum_{k\in\mathcal{K}_i} v_i^{\text{a}} \leq \alpha\sum_k 1 \qquad (\because \textstyle\sum_{j\in\mathcal{J}}\Theta_{kj} = 1\ \forall k \text{ by definition})$$

$$\sum_{i\in\mathcal{I}} n_i v_i^{\text{a}} \leq |\mathcal{K}|\alpha$$

$$\alpha \geq \frac{\sum_{i\in\mathcal{I}} n_i v_i^{\text{a}}}{|\mathcal{K}|}$$

Note that the R.H.S. above is a constant which bounds the variable α. Hence, this constraint facilitates early pruning of infeasible solutions, leading to a substantial speedup.

5 Experimental Results

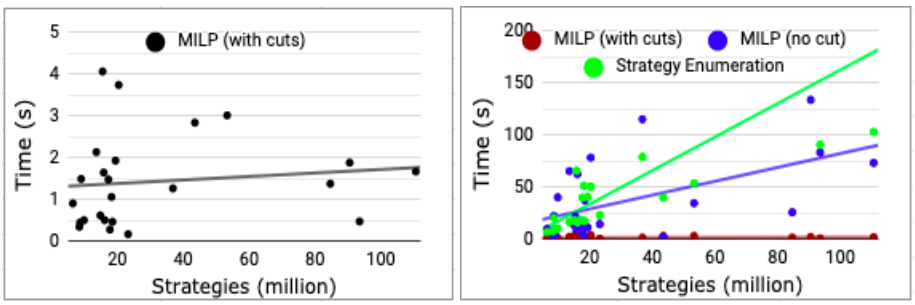
For runtime comparisons, we consider relatively small problem sizes, since they suffice to clearly highlight the efficiency of our solution methods, which is very crucial when scaling to large-scale enterprise networks.

Setup. We vary the parameters $|\mathcal{K}|$, $|\mathcal{I}|$, $|\mathcal{J}|$, where a particular triple of $|\mathcal{K}|$, $|\mathcal{I}|$, $|\mathcal{J}|$ defines a "scenario" for an experiment. Within each "scenario" we consider various "settings" by randomly assigning TCs to machines, and sampling feasibility constraints Π_{ij}, costs c_{ij} and budget B. Within each "setting" , we consider various "instances" by randomly sampling valuations v^{d}, v^{a}. For any scenario considered, we report averaged results over 30 problem settings, and 30 instances from each setting.

We now evaluate the runtime and the solution quality of our solutions, starting with the more general constrained CCG.

5.1 Constrained CCG

MILP Runtime analysis. We compare the runtime of MILP with and without cut-augmentation against a benchmark which computes the optimal strategy by obtaining the normal-form game via explicit strategy enumeration. We set $|\mathcal{K}| = 30$, and $|\mathcal{I}| = |\mathcal{J}| = 6$ and randomly sample 30 settings with strategy space size between 5 and 125 million. For each setting, we compute the average runtimes over 30 instances. The runtime analysis is presented in Fig. 2 (the individual instances are shown as the points with the line showing the overall linear trend). Fig. 2a shows that the MILP when augmented with can be computed within 4 seconds for instances with upto 125 million. In comparison, Fig. 2b shows that the strategy enumeration approach is computationally very expensive and grows linearly with the number of strategies (which grows exponentially in game parameters). The runtime of MILP without cuts is also seen to grow, although slower. However, the runtime of MILP when augmented with cuts is markedly lower, providing up to 100-fold speedup over the benchmark. Table 1 shows the value of the two cuts separately; the symmetry breaking cuts independently improve runtime by up to 7 times, while the attacker valuation bounding cut produces up to 30 times speedup.



(a) Runtime of MILP with cuts against the strategy space size

(b) Comparison of MILP with and without cuts, and Strategy enumeration benchmark

Fig. 2: Runtime Analysis

Parameters ($\|\mathcal{K}\|$, $\|\mathcal{I}\|$, $\|\mathcal{J}\|$)	No cuts	Symmetry breaking	Attacker val. bound	Both
20,4,4	0.42	0.21	0.11	0.10
25,5,5	17.45	1.50	0.50	0.39
30,6,6	49.59	7.21	1.71	1.45

Table 1: MILP runtime (sec): Cut-augmentation impact

Parameters ($\|\mathcal{K}\|$, $\|\mathcal{I}\|$)	2 OCs	3 OCs	4 OCs
200, 10	10.1	114	138
200, 25	10.2	78	121

Table 2: MILP runtime (sec) when 2 OCs can mask all TCs

5.2 Unconstrained CCG

Optimality versus Efficiency. We compare the Runtime of the MILP which computes the optimal solution, against the FPTAS for various values of ϵ (levels of approximation). Table 3a shows that the FPTAS runtime rapidly increases with ϵ as expected. In comparison to the MILP, for $\epsilon = 0.01$, it is 1.5 to 2.7 times as fast as the MILP across the 4 settings, while allowing lower precision with $\epsilon = 0.1$ leads to 20-50 times improvement in runtime. Table 3b shows that the actual gap between the solution computed by the FPTAS algorithm and the optimal solution is two to four times smaller than the set precision ϵ.

Parameters ($\|\mathcal{K}\|$, $\|\mathcal{I}\|$)	MILP runtime	FPTAS runtime		
		$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.01$
200, 25	11.45	0.20	0.42	2.9
500, 25	71.14	2.65	5.58	30.4
200, 50	8.72	0.24	0.50	3.1
500, 50	43.72	2.98	6.02	31.2

(a) Runtime(sec) of MILP and FPTAS: varying ϵ.

Parameters ($\|\mathcal{K}\|$, $\|\mathcal{I}\|$)	FPTAS optimality gap		
	$\epsilon = 0.1$	$\epsilon = 0.05$	$\epsilon = 0.01$
200, 25	0.044	0.014	0.002
500, 25	0.029	0.013	0.002
200, 50	0.055	0.017	0.003
500, 50	0.068	0.023	0.004

(b) FPTAS solution quality

Table 3: FPTAS runtime and solution quality

MILP speedup with restriction to 2 OCs. For $|\mathcal{K}| = 200$, and $|\mathcal{I}| \in \{10, 25\}$, we consider cases with 3 or 4 OCs of which 2 can mask all TCs at no cost. For comparison, we consider the MILP runtime which only these 2 OCs were considered. For these scenarios, it can be seen from Table 2 that considering only 2 OCs which can mask all the machines, leads to an average runtime that is 8-12 times lower, compared to having all OCs. This shows that pre-processing of the input to find such two OCs proves to cause a huge reduction in runtime.

6 Summary

In this paper, we present a general-sum Cyber Deception Game model for strategic cyber deception, also addressing uncertainties in the defender's knowledge of attacker valuations. We first consider the *unconstrained* CDG where there are no constraints

on the feasibility or cost of deception strategies and prove that computing an optimal solution is NP-hard here, in contrast to the zero-sum counterpart where it is to simply mask all machines with the same OC. This result highlights the challenge in optimizing the strategy even when practical logistical or technological constraints in deploying deception are overcome. We also present a Fully Polynomial Time Approximation Scheme (FPTAS) for the aforementioned NP-hard problem. For CDG with constraints, we present an MILP formulation to find the optimal solution, boosted with computational improvements using compact strategy representations and added constraints. Finally, we experimentally evaluate our algorithms and show that they give substantial improvements in scalability.

7 Acknowledgements

This research was sponsored by the Army Research Office (grant W911NF-17-1-0370) and also in part by National Science Foundation (grant IIS-1850477) and Army Reserch Lab's Cyber Security CRA (grant W911NF-13-2-00).

Appendix

Complete MILP formulation for OP (2)
We let $\underline{v}^{\mathrm{d}}$, $\overline{v}^{\mathrm{d}}$ denote the least and the highest defender valuations, and similarly, $\underline{v}^{\mathrm{a}}$, $\overline{v}^{\mathrm{a}}$ the least and the highest attacker valuations. To linearize, we let the variables X_{kj}, Y_{kj}, and Z_{kj} represent the bilinear terms $(1-q_j)\Theta_{kj}$, $\alpha\Theta_{kj}$, and $\gamma\Theta_{kj}$ respectively and add liner constraints which enforce the appropriate product value to them. The resultant MILP is as follows.

$$
\begin{aligned}
\max_{\Phi,\boldsymbol{q},\gamma,\alpha} \quad & \gamma \\
\text{s.t.} \quad & \alpha,\gamma \in \mathbb{R},\ \Phi \in \mathcal{F},\ \boldsymbol{q} \in \{0,1\}^{|\mathcal{J}|} \\
& \sum_{j\in\mathcal{J}} q_j \geq 1 \\
& \epsilon \sum_k X_{kj} \leq \sum_k Y_{kj} - \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} \Theta_{kj} v_i^{\mathrm{a}} && \forall j \in \mathcal{J} \\
& \sum_k Y_{kj} - \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} \Theta_{kj} v_i^{\mathrm{a}} \leq \sum_k X_{kj} M && \forall j \in \mathcal{J} \\
& \sum_k X_{kj} M + \sum_{i\in\mathcal{I}} \sum_{k\in\mathcal{K}_i} \Theta_{kj} v_i^{\mathrm{d}} \geq \sum_k Z_{kj} && \forall j \in \mathcal{J} \\
& q_j \leq \sum_{k\in\mathcal{K}} \Theta_{kj} && \forall j \in \mathcal{J} \\
& X_{kj} + q_j \leq 1 && \forall k \in \mathcal{K}\ \forall j \in \mathcal{J} \\
& X_{kj} \leq q_j && \forall k \in \mathcal{K}\ \forall j \in \mathcal{J} \\
& X_{kj} + q_j \geq \Theta_{kj} && \forall k \in \mathcal{K}\ \forall j \in \mathcal{J}
\end{aligned}
$$

$$\underline{v}^{\mathrm{a}}\Theta_{kj} \leq Y_{kj} \leq \overline{v}^{\mathrm{a}}\Theta_{kj} \qquad \forall k \in \mathcal{K}\ \ \forall j \in \mathcal{J}$$
$$\alpha + \overline{v}^{\mathrm{a}}\Theta_{kj} \leq \overline{v}^{\mathrm{a}} + Y_{kj} \qquad \forall k \in \mathcal{K}\ \ \forall j \in \mathcal{J}$$
$$\alpha + \underline{v}^{\mathrm{a}}\Theta_{kj} \geq \underline{v}^{\mathrm{a}} + Y_{kj} \qquad \forall k \in \mathcal{K}\ \ \forall j \in \mathcal{J}$$
$$\underline{v}^{\mathrm{d}}\Theta_{kj} \leq Z_{kj} \leq \overline{v}^{\mathrm{d}}\Theta_{kj} \qquad \forall k \in \mathcal{K}\ \ \forall j \in \mathcal{J}$$
$$\gamma + \overline{v}^{\mathrm{d}}\Theta_{kj} \leq \overline{v}^{\mathrm{d}} + Z_{kj} \qquad \forall k \in \mathcal{K}\ \ \forall j \in \mathcal{J}$$
$$\gamma + \underline{v}^{\mathrm{d}}\Theta_{kj} \geq \underline{v}^{\mathrm{d}} + Z_{kj} \qquad \forall k \in \mathcal{K}\ \ \forall j \in \mathcal{J}$$

Bibliography

Achleitner S, La Porta T, McDaniel P, Sugrim S, Krishnamurthy SV, Chadha R (2016) Cyber deception: Virtual networks to defend insider reconnaissance. In: Proceedings of the 8th ACM CCS international workshop on managing insider threats, ACM, pp 57–68

Albanese M, Battista E, Jajodia S, Casola V (2014) Manipulating the attacker's view of a system's attack surface. In: Communications and Network Security (CNS), 2014 IEEE Conference on, IEEE, pp 472–480

Albanese M, Battista E, Jajodia S (2016) Deceiving attackers by creating a virtual attack surface. In: Cyber Deception, Springer, pp 169–201

Alpcan T, Başar T (2010) Network security: A decision and game-theoretic approach

Arkin O, Yarochkin F (2003) A fuzzy approach to remote active operating system fingerprinting. URL http://www.syssecurity.com/archive/papers/Xprobe2.pdf

Auffret P (2010) Sinfp, unification of active and passive operating system fingerprinting. Journal in Computer Virology 6(3):197–205, DOI 10.1007/s11416-008-0107-z, URL https://doi.org/10.1007/s11416-008-0107-z

Berrueta DB (2003) A practical approach for defeating nmap os- fingerprinting

Breton M, Alj A, Haurie A (1988) Sequential stackelberg equilibria in two-person games. Journal of Optimization Theory and Applications DOI 10.1007/BF00939867, URL https://doi.org/10.1007/BF00939867

Chadha R, Bowen T, Chiang CJ, Gottlieb YM, Poylisher A, Sapello A, Serban C, Sugrim S, Walther G, Marvel LM, Newcomb EA, Santos J (2016) Cybervan: A cyber security virtual assured network testbed. In: MILCOM 2016 - 2016 IEEE Military Communications Conference, DOI 10.1109/MILCOM.2016.7795481

De Gaspari F, Jajodia S, Mancini LV, Panico A (2016) Ahead: A new architecture for active defense. In: Proceedings of the 2016 ACM Workshop on Automated Decision Making for Active Cyber Defense

Durkota K, Lisỳ V, Bosanskỳ B, Kiekintveld C (2015) Optimal network security hardening using attack graph games. In: IJCAI

Ferguson-Walter K, LaFon D, Shade T (2017) Friend or faux: Deception for cyber defense. Journal of Information Warfare

Goel V, Perlroth N (2016) Yahoo Says 1 Billion User Accounts Were Hacked. https://www.nytimes.com/2016/12/14/technology/yahoo-hack.html

Guo Q, Gan J, Fang F, Tran-Thanh L, Tambe M, An B (2018) On the inducibility of stackelberg equilibrium for security games. CoRR abs/1811.03823

Gutzmer I (2017) Equifax Announces Cybersecurity Incident Involving Consumer Information. https://investor.equifax.com/news-and-events/news/2017/09-07-2017-213000628

Jiang AX, Chan H, Leyton-Brown K (2017) Resource graph games: A compact representation for games with structured strategy spaces. In: AAAI

Joyce R (2016) Disrupting nation state hackers. USENIX Association, San Francisco, CA

Kiekintveld C, Marecki J, Tambe M (2011) Approximation methods for infinite bayesian stackelberg games: Modeling distributional payoff uncertainty. In: AAMAS, URL http://dl.acm.org/citation.cfm?id=2034396.2034412

Kiekintveld C, Islam T, Kreinovich V (2013) Security games with interval uncertainty. In: AAMAS

Laszka A, Vorobeychik Y, Koutsoukos XD (2015) Optimal personalized filtering against spear-phishing attacks. In: AAAI

Lewis J (2018) Economic Impact of Cybercrime. https://www.csis.org/analysis/economic-impact-cybercrime

Lyon GF (2009) Nmap network scanning: The official Nmap project guide to network discovery and security scanning

MacFarland DC, Shue CA (2015) The sdn shuffle: creating a moving-target defense using host-based software-defined networking. In: Proceedings of the Second ACM Workshop on Moving Target Defense, ACM, pp 37–41

Mandiant (2013) Apt1: Exposing one of china's cyber espionage units

Nguyen TH, Yadav A, An B, Tambe M, Boutilier C (2014) Regret-based optimization and preference elicitation for stackelberg security games with uncertainty. In: AAAI, URL http://dl.acm.org/citation.cfm?id=2893873.2893991

Peterson A (2015) OPM says 5.6 million fingerprints stolen in cyberattack, five times as many as previously thought. https://www.washingtonpost.com/news/the-switch/wp/2015/09/23/opm-now-says-more-than-five-million-fingerprints-compromised-in-breaches

Pıbil R, Lisỳ V, Kiekintveld C, Bošanskỳ B, Pechoucek M (2012) Game theoretic model of strategic honeypot selection in computer networks. Decision and Game Theory for Security

Pita J, Jain M, Tambe M, Ordóñez F, Kraus S (2010) Robust solutions to stackelberg games. Artif Intell 174(15):1142–1171, DOI 10.1016/j.artint.2010.07.002, URL http://dx.doi.org/10.1016/j.artint.2010.07.002

Rahman MA, Manshaei MH, Al-Shaer E (2013) A game-theoretic approach for deceiving remote operating system fingerprinting. 2013 IEEE Conference on Communications and Network Security (CNS) pp 73–81

Schlenker A, Xu H, Guirguis M, Kiekintveld C, Sinha A, Tambe M, Sonya S, Balderas D, Dunstatter N (2017) Don't bury your head in warnings: A game-theoretic approach for intelligent allocation of cyber-security alerts

Schlenker A, Thakoor O, Xu H, Fang F, Tambe M, Tran-Thanh L, Vayanos P, Vorobeychik Y (2018) Deceiving cyber adversaries: A game theoretic approach. In: AAMAS, URL http://dl.acm.org/citation.cfm?id=3237383.3237833

Serra E, Jajodia S, Pugliese A, Rullo A, Subrahmanian V (2015) Pareto-optimal adversarial defense of enterprise systems. ACM Transactions on Information and System Security (TISSEC) 17(3):11

Sinha A, Malo P, Deb K (2018) A review on bilevel optimization: From classical to evolutionary approaches and applications. IEEE Transactions on Evolutionary Computation 22(2):276–295, DOI 10.1109/TEVC.2017.2712906

von Stengel B, Zamir S (2004) Leadership with commitment to mixed strategies. Tech. rep.

Tambe M (2011) Security and game theory: algorithms, deployed systems, lessons learned

Thinkst (2015) Canary. https://canary.tools/

Tijs SH (1981) Nash equilibria for noncooperative n-person games in normal form. SIAM Review URL http://www.jstor.org/stable/2029993