Budapest University of Technology and Economics
Department of Networked Systems and Services

Robustness Against Strategic Attacks

Ph.D. Dissertation

of

Áron Lászka

Supervisor:

Levente Buttyán, Ph.D.

Budapest, 2014

Contents

List of Figures

Chapter 1

Introduction

Traditionally, security research has primarily been focused on preventing adversaries from carrying out successful attacks against a network or a system. However, in many situations, attaining this goal is either economically infeasible or impossible. For example, physical networks are inherently vulnerable to denial-of-service type attacks, such as physical node destruction and jamming. Thwarting all of these attacks would entail protecting every single element of a network, which would require substantial investments from a defender. In many networks, this is not economically feasible. As another motivation, consider recent examples from the world of cyber-warfare, which have shown that even secluded and highly protected computer systems can be penetrated by a determined and resourceful attacker. For instance, the Stuxnet worm was able to penetrate highly protected computer systems that were not connected to the Internet, such as nuclear facilities [Kaspersky Lab, 2010].

In these scenarios, where perfect protection is not a realistic goal, the defender has to resort to mitigating the effects of successful attacks. In other words, the defender's goal in these scenarios needs to be ensuring that she will sustain only tolerable losses in the event of a successful attack. This goal can be achieved either proactively by designing the network or systems to be robust, that is, resilient to attacks; or reactively by using it in an attack-resilient manner. In this dissertation, I focus on strategic attacks, in which an adversary targets a particular target and tailors her attack to it. The strategic nature of the attacks will pose certain challenges, which I will discuss shortly.

In this dissertation, I study four problems related to robustness against strategic attacks. Note that each problem will be introduced in more detail in the corresponding chapter.

- First, I study the robustness of network topologies against strategic attacks. The main motivation for this research is that, in order to design robust networks, one first has to be able to quantify robustness. Although one can find a number of metrics in the literature that can be used to quantify robustness, the majority of previous metrics are not based on specific attacker and network models or disregard the strategic interactions between an attacker and a defender. In contrast, I adopt a game-theoretic approach recently proposed in [Gueye et al., 2010], which derives metrics from given attacker and network communication models. This approach allows one to find the right metric for a given application scenario by finding the attacker and network communication models that are the most appropriate for the scenario. In this dissertation, I focus on providing computational complexity results for this approach, proposing novel network communication models, and generalizing the approach to consider additional technical and economic constraints on the network.

- Second, I study the design of robust wireless sensor networks. A sensor network consists of a large number of spatially distributed sensor nodes, which measure their environment and forward measurement data through the network to a data collection center, called the sink node. Sensor networks are envisioned to have many applications, including military applications, such as battlefield surveillance, and critical infrastructure protection, such as surveillance of electric power networks. However, wireless sensor networks (WSNs) are usually assumed to consist of physically unprotected nodes and links, which makes them easy targets for denial-of-service type attacks. One of the key elements of designing networks that are resilient to attacks is finding a robust topology.

In the case of WSNs, the topology of a network is determined by the placement of the nodes. Since the placement of the sensor nodes is usually given by the application, in this dissertation, I focus on the problem of robust sink node placement.

- Third, I study mitigation strategies against covert compromises. Attackers of computing resources often aim to keep security compromises hidden from the defenders in order to extract more value over a longer period of time. For example, in cyber-espionage, an attacker who has compromised an account will try to remain covert in order to be able to spy on the user as long as possible. If detecting or preventing these covert compromises is too expensive for the defender, the effects of the compromises need to be mitigated by moving the resource into a known secure state (e.g., by changing the password of a potentially compromised account). However, since the attacks are stealthy, the defender has to schedule mitigation moves without knowing when a move will actually be useful. In this dissertation, I focus on finding strategies for scheduling mitigation moves that are resilient to attacks.

- Fourth, I study bribe-resilient team composition in organizations. In cyber-espionage, an adversary can try to sidestep the technical security mechanisms of an organization (or a company) by having an employee bribed or compromised using social-engineering. However, if the secret sought by the adversary needs to be shared only with a subset of the employees, a manager can limit the success probability of the attacks by using a randomized sharing strategy. More specifically, if a manager chooses the subset of employees, with whom the secret is shared, in a random and privy way, then the adversary has to select her target without knowing if it will be of any value to her. Hence, a manager can mitigate the effects of attacks sidestepping technical security by using randomized team composition, which forces the adversary to attack blindly. In this dissertation, I focus on finding team composition (i.e., secret sharing) strategies that are resilient to strategic attacks.

Even though attack-resilience may resemble tolerance to random faults at first glance, they are actually fundamentally different. The main difference between the two notions is the assumption that the occurrence of random faults does not depend on the actions of the defender, while strategic attacks do. More specifically, in the case of random-fault-tolerance, one can usually assume that the elements of a system fail with given probabilities, independently of each other. Hence, one can find the best defense by minimizing losses based on an a priori given fault distribution. In the case of strategic-attack-resilience, on the other hand, the adversary anticipates the defender's actions and chooses the attack that works best against the actions taken by the defender. Consequently, one has to find the best defense by minimizing losses based on what the attacker might do against a given defense.

These strategic interactions between the defender and the attacker are modeled most naturally using the game theory nomenclature. Game theory is the mathematical study of conflict and cooperation between strategic decision-makers [Myerson, 1991], who are called the players of a game. Resilience against strategic attacks can be studied using attacker-defender games, where one or more players take the role of strategic adversaries, and one player takes the role of the defender. In this dissertation, I model the first problem as a two-player game between a network operator and a strategic adversary capable of removing the elements of the network; the third problem as a game between a defender capable of performing mitigation moves and a stealthy strategic attacker; and the fourth problem as a game between a manager responsible for team composition and a spy who can bribe employees. The analysis of the resulting games will allow us to gain insight into each of these resilience problems.

Since the adversary and the defender can both anticipate their opponents' actions, attack-resilience problems can be more complicated than random-fault-tolerance problems. As I mentioned before, in the latter, one usually has to minimize losses based on a priori given fault probabilities, while in the latter, the objective function to be minimized can be more complex. Moreover, in the four problem studied in this dissertation, the cardinalities of the sets of possible designs or strategies are generally exponential (or even greater than that) in the size of the input, which presents further computational challenges. For example, in the first and and the fourth problems, the cardinality of the set of strategies available to the network operator and the manager, respectively, are exponential in the size of the input (i.e., in the size of a description of the network or a list of employees). As another example, the cardinalities of the set of possible placements in the second problem and the set of strategies available to the defender in the third problem are greater even than continuum. Consequently, we cannot solve any of these problems using a simple exhaustive search. In this dissertation, I will show that some of the problems can actually be solved efficiently, while other problems are computationally intractable.

To show which problems can be solved efficiently and which problems are intractable, I use the theory of computational complexity. Computational complexity theory classifies computational problems based on their inherent difficulty [Garey and Johnson, 1979]. The two complexity classes that will play a significant role in this dissertation are P and NP. The first class, P, consists of all problems that can be solved in polynomial time. This class is important because – according to the widely accepted Cobham-Edmonds thesis [Cobham, 1965]– these problems can be feasibly computed on some computational device. Hence, given a computational problem, my primary goal will be to find a polynomial-time algorithm for solving it, and I will refer to such algorithms as efficient algorithms. The second significant class, NP, consists of all problems whose solutions can be verified in polynomial time. This class is important because it is a widely accepted conjecture that P $\neq$ NP. In other words, it is a widely accepted conjecture that there are problems in NP that cannot be solved in polynomial time. Consequently, if we can prove that a problem is at least as hard as the hardest problem in NP, we can claim that the problem is computationally intractable. We call such problems NP-hard.

Unfortunately, robust design presents other challenges besides computational complexity: robustness often comes at a price. Consider – for example – the third problem, where a defender faces covert attacks, against which she can defend herself only by making mitigation moves. In this case, the more often the defender makes a mitigation move, the more resilient her strategy is against covert compromises. However, these mitigation moves entail some cost, which might be negligible at first, but as the frequency of the moves increases, it can easily attain a value that is comparable to the benefits of increased resilience. Therefore, an economically rational defender has to find the right balance between minimizing costs and maximizing robustness. More generally, finding the right strategy or design is often a trade-off problem between costs and resilience. To study this aspect of robustness, I introduce a network link usage based cost model in the first problem, sink node placement costs in the second problem, and mitigation move costs in the third problem.

The remainder of this dissertation is organized as follows. First, in Section 1.1, I summarize the notations and terminology used throughout this dissertation. In Chapter 2, I study the robustness of network topologies using a game-theoretic model. The results presented in this chapter have been published in [Laszka et al., 2012b, Laszka et al., 2012c, Laszka and Gueye, 2013a, Laszka and Gueye, 2013b]. In Chapter 3, I study the design of wireless sensor network topologies that are resilient to strategic attacks. All results of this chapter have been published in [Laszka et al., 2011, Laszka et al., 2013a]. In Chapter 4, I study mitigation strategies against covert compromises of computing resources. All results of this chapter have been published in [Laszka et al., 2013c, Laszka et al., 2013d]. In Chapter 5, I study bribe-resilient team-composition strategies. The results presented in this chapter have been published in [Laszka et al., 2013e]. Finally, in Chapter 6, I conclude the paper by summarizing the main results of each chapter, comparing them to each other, and listing possible applications.

1.1 Notation and Terminology

Note that the purpose of this section is only to establish the notation and terminology used throughout this dissertation, it is not meant to be a standalone introduction to any of the covered topics.

The set of real numbers is denoted by $\mathbb{R}$, the set of non-negative real numbers is denoted by $\mathbb{R}_{\geq 0}$, and the set of positive real numbers is denoted by $\mathbb{R}^+$. The set of integers is denoted by $\mathbb{Z}$, and the set of natural numbers is denoted by $\mathbb{N}$. Note that the set of natural numbers includes 0 (i.e., $\mathbb{N} = \{0, 1, 2, \ldots\}$).

I use $\cdot$ to denote multiplication explicitly in some formulas, which would be less readable otherwise (e.g., $S \cdot y$ instead of Sy).

I use Leibniz's notation $\frac{d}{dx} f(x)$ to denote the derivative of function $f(x)$ with respect to x. I use the standard notation $\int_a^b f(x)dx$ to denote the definite integral of function $f(x)$ over the interval $[a, b]$.

Finally, I define the function

$$1_{condition} = \begin{cases} 1 & \text{if } \textit{condition} \text{ is true} \\ 0 & \text{otherwise.} \end{cases} \tag{1.1}$$

Vectors and Matrices

Vectors are assumed to be column vectors and denoted by bold lowercase letters (e.g., $\boldsymbol{x}, \boldsymbol{\alpha}$). Vectors of ones and zeros are denoted by $\mathbf{1}$ and $\mathbf{0}$, respectively (their sizes are not indicated, as they are never

ambiguous in this dissertation). Matrices are denoted by bold uppercase letters (e.g., $\boldsymbol{A}, \boldsymbol{\Lambda}$). I use the prime sign to denote transposition (e.g., $\boldsymbol{x}'$, $\boldsymbol{\Lambda}'$), and subindices (e.g., x_e, α_T) to refer to elements of vectors (e.g., $\boldsymbol{x} = [x_1, \ldots, x_n]'$). The support of a vector $\boldsymbol{x}$ is the set of the indices of its positive elements; formally, the support of $\boldsymbol{x}$ is $\{i \mid x_i > 0\}$. As an example to using these notations, consider the equality $\mathbf{1}'\boldsymbol{x} = \sum_i x_i$, which I will use repeatedly throughout this dissertation.

I call a geometric object with flat sides, which can exist in any general number of dimensions, a polyhedron. Note that the term polyhedron is often used to denote only objects in three dimensions. I use the term polyhedron in the more general sense following related work [Fulkerson, 1971] and [Gueye, 2011].

Random Variables and Probability Distributions

Random variables are denoted by uppercase letters (e.g., T, B), probability distributions are denoted by uppercase calligraphic letters (e.g., $\mathcal{T}$,$\mathcal{U}$), and the probability of an event is denoted by Pr[event] (e.g., the probability that the realization of T will be at most b is $\Pr[T \leq b]$). The cumulative distribution function of a random variable X is denoted by $F_X(x) = \Pr[X \leq x]$, while its probability density function is a measurable function f_X with the property that $\Pr[X \in A] = \int_A f \, d\mu$ for any measurable set A.

Graphs

Let $G = (\mathcal{V}, \mathcal{E})$ be a graph, where $\mathcal{V}$ is a set of vertices (or nodes) and $\mathcal{E}$ is a set of edges (or links). I use the term vertices interchangeably with the term nodes, and edges interchangeably with links. In Chapter 2, I will prefer the term node to avoid confusion with the vertices of a polyhedron (i.e., its extreme points). In directed graphs, the edges are also called arcs. Again, I use the terms edge, link, and arc interchangeably. In Chapter 3, where many discussions use multiple graphs at the same time, I use $\mathcal{V}(G)$ and $\mathcal{E}(G)$ to denote the vertex and edge set of a graph G explicitly. I let $G[T \setminus \{e\}]$ denote the graph $G' = (\mathcal{V}, T \setminus \{e\})$. Finally, I use vertices and edges as subindices to refer to elements of vectors of length $|\mathcal{V}|$ and length $|\mathcal{E}|$ (e.g., for a $\boldsymbol{y} \in \mathbb{R}^{|\mathcal{E}|}$ and an $e = (u, v) \in \mathcal{E}$, I use y_e or $y_{(u,v)}$ by implicitly assuming that the edges have been numbered).

Network Flows For a directed graph $G = (\mathcal{V}, \mathcal{E})$, a flow is a function $f : \mathcal{E} \mapsto \mathbb{R}_{\geq 0}$, while an integer flow is a function $f : \mathcal{E} \mapsto \mathbb{N}$. For an edge $e = (u, v)$, the flow $f((u, v))$ (or $f(e)$) along the edge is the amount transported from u to v. In a feasible flow, for every node v, the net outgoing flow $\sum_{(v,w)\in\mathcal{E}} f((v, w)) - \sum_{(u,v)\in\mathcal{E}} f((u, v))$ has to be zero, unless v is a source (or a sink) node, which can produce (or consume) flow.

In undirected graphs, an edges can transport flow in both directions. For an edge $\{u, v\}$, I let $f(u, v) \in \mathbb{R}_{\geq 0}$ denote the amount transported from u to v, and $f(v, u) \in \mathbb{R}_{\geq 0}$ denote the amount transported from v to u. Hence, the net outgoing flow of node v in an undirected graph is $\sum_{(u,v)\in\mathcal{E}} f(v, u) - f(u, v)$. Note the lack of double parentheses, which differentiates directed and undirected network flows in notation.

Game Theory

The decision makers of a game are called the players. Each player has to choose from one or more options, which are called strategies. If some players choose their strategies before others, who will have some information regarding the choices of the former, then the game is called sequential; otherwise, it is called simultaneous. The set of all strategies available to a player is called the player's strategy set. The utility of a given outcome for a player is called the player's payoff, while the additive inverse of the payoff is called the player's loss.

A strategy profile describes how each player will choose, that is, it assigns exactly one strategy to each player. A profile is a Nash equilibrium if no player can increase her payoff by changing her strategy unilaterally. A profile is a subgame-perfect equilibrium if it represents a Nash equilibrium of every subgame. In this dissertation, I use the term equilibria to refer to the Nash equilibria of a game.

Computational Complexity

A decision problem is a formal question with a yes-or-no answer. The class P contains all decision problems that can be solved by a deterministic Turing machine in polynomial time. The class NP contains

all decision problems where a positive answer always has some proof that is verifiable in polynomial time by a deterministic Turing machine. Finally, the class NP-hard contains decision problems to which every problem in NP is reducible. Intuitively, reduction from one problem to another means proving that the former is a special case of the latter. An optimization problem is the problem of finding the best solution from the set of feasible solutions. For every optimization problem, there is a corresponding decision problem. In this dissertation, when I informally refer to the computational complexity of an optimization problem, I implicitly refer to the complexity of a decision version that is of roughly equal computational difficulty.

Chapter 2

Robustness of Network Topologies

2.1 Introduction

As our dependence on networks increases, so does the need to be able to protect them from malicious attacks. For example, many critical infrastructures depend on networks, which makes them ideal targets for terrorist and other adversaries. Ideally, one would prevent these attacks from happening at all; however, this is often impossible or economically infeasible. For example, consider a wireless sensor network whose nodes are distributed over a large area. To protect the whole network, one would have to make substantial investments into protecting each node from physical destruction, wireless jamming, etc.; however, wireless sensor networks are envisioned to be relatively cheap. Since we cannot prevent all possible attacks, we have to design networks that retain most of their functionality even after a successful attack. In other words, we have to design robust networks.[1] The robustness of a network depends on many factors; in this chapter, I study the robustness of network topologies.[2]

In order to be able to design robust network topologies, which are resilient to strategic attacks, one must first be able to quantify the robustness of topologies.[3] Quantifying the robustness – or equivalently, the vulnerability – of topologies has been extensively studied, for example, in [Cunningham, 1985, Holme et al., 2002, Grubesic et al., 2008]. A more detailed discussion of some of the previously proposed metrics will follow in Section 2.7. Unfortunately, the simultaneous and strategic decision making of the network operator and the adversary, which is of key importance when modeling strategic attacks, has received only little attention.

Recently, however, Gueye et al. proposed another approach for quantifying the robustness of network topologies in a series of papers [Gueye et al., 2010, Gueye et al., 2011, Gueye et al., 2012, Gueye and Marbukh, 2012]. In their approach, the strategic interaction between an adversary and the operator of a network are modeled as an attacker-defender game, called a *network blocking game* (NBG). An NBG is defined by the network topology, whose robustness is to be studied, and a communication model, which describes how the network is to be used by the operator. After solving the game, its Nash equilibrium strategies are used to predict the adversary's most likely actions, while the adversary's equilibrium payoff[4] serves as a metric for the vulnerability (i.e., inverse robustness) of the network. The idea behind this approach is that, if the adversary's equilibrium payoff is high, then the network is easy to attack; hence, it is vulnerable to attacks mounted by a strategic adversary. On the other hand, if the adversary's equilibrium payoff is low, then the network is robust against strategic attacks.

With respect to computational complexity, network blocking games present two very interesting

[1] Note that, even though robustness against intentional attacks resembles tolerance against random faults in many respects, they are fundamentally different: the latter assumes some given distribution of faults, while the former assumes a strategic adversary, who anticipates defense and considers the defender's strategy when she decides where to attack.

[2] By topology, I mean the logical layout of the network nodes and the connections between them, which can be represented mathematically using a graph.

[3] By the robustness of a network, I mean the extent to which the network retains its functionality after a successful attack. Note that I do not give a more formal definition of robustness at this point, since that is actually one of this chapter's goals. In other words, one of this chapter's goals is to be able to tell how robust a network is based on some assumptions on the adversary and the network.

[4] It has been shown that the adversary's payoff is the same in every equilibrium of a network blocking game.

challenges. First, in most communication models, the network operator's pure-strategy set is generally exponential in the size of the network. Second, the input of the computation problem is not the game itself, but the topology of the network. Hence, traditional methods for solving two-player games, which assume that the payoff matrix – or some equivalent definition of the players' strategies and payoffs – is given, cannot be applied in practice, as computing the payoff matrix alone would require exponential running time. Consequently, I will pay close attention to the computational complexity of solving network blocking games both in general and in particular (classes of) communication models.

The outline of the remainder of this chapter is the following. First, in Section 2.2, I introduce the general framework of network blocking games and summarize the previous results on which my analysis builds. Then, in Section 2.3, I present complexity-theory results on solving network blocking games in general. More specifically, I show that the problem is generally NP-hard, it but can be solved efficiently for an important subclass of communication models. In Section 2.4, I propose a novel communication model, called the All-to-One model. I show that a network blocking game in the All-to-One model can be solved efficiently and that the resulting robustness metric is closely related to a previously proposed metric, called directed graph strength. In Section 2.5, I propose another communication model, called the All-to-All model with linear usage (or loss). I show that a network blocking game in the All-to-All model with linear usage can be solved efficiently and that the resulting metric is closely related to a previously proposed metric, called the Cheeger constant. Next, in Section 2.6, I extend the general framework of network blocking games by introducing a budget constraint on the operator. I propose two budget constraint formulations, called the Maximum Cost and the Expected Cost Constraints. I show that former leads to NP-hard computational problems, while the latter leads to problem which can be solved efficiently. Finally, in Section 2.7, I review related work on the robustness of network topologies, and in Section 2.8, I conclude the chapter.

2.2 Network Blocking Games

In this section, I introduce the general framework of network blocking games and – as an illustration – one of the previously proposed communication models. Then, I summarize the results of [Gueye, 2011] characterizing the equilibria of the game, on which my analysis presented in the subsequent sections builds.

A network blocking game is defined by a network topology, a communication model, and an attacker model. The topology of a network is represented by a connected graph $G = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ is the set of nodes and $\mathcal{E}$ is the set of links. The nodes can model network nodes, such as routers, switches, host computers, and terminal equipments, while the links (also called edges) can model connections between the nodes, such as wired or wireless links. The links (and the graph) can be either undirected or directed depending on the communication model (as we will see later). Note that we can model both physical and logical topologies; however, I am primarily interested in studying the robustness of physical topologies.

The network operator's goal is to provide some connectivity between the nodes. To achieve this, she selects a collection T of the network elements to be used for communication. Note that I use the term collection in a broad sense, that is, it is not simply a subset of network elements but rather a configuration of how they are to be used. For example, in the Supply-Demand model (which will be discussed in more detail shortly), a collection is an integer flow. The type of connectivity that has to be provided and the set of feasible collections that the operator can use, denoted by $\mathcal{T}$, are determined by the communication model. Note that we assume that there always exists at least one feasible collection; otherwise, the operator would not be able to use the network as intended (even without the presence of an adversary).

The adversary's goal is to disrupt the communication between the nodes. To achieve this, she targets a network element, which is removed from the graph (or disabled) as a result of her attack. Attacking an element usually models some denial-of-service type attack, such as physical destruction or wireless jamming, but more generally, it could also model compromising an element and eavesdropping on the traffic traversing it. In this dissertation, I will usually assume that the set of network elements that can be targeted by the adversary is the set of edges $\mathcal{E}$. In other words, the attacker model usually assumes that adversary is capable of removing only edges. However, for most communication models, the results for a more general attacker model, which allows attacking both nodes and edges, can be derived easily using vertex splitting. In Section 2.4.2, I show how this can be done in the case of the All-to-One

communication model.

For a given collection T and link e, let $\lambda(T, e)$ denote how much link e is used when the operator is employing collection T. This usage value can model, for example, the amount of traffic on e or the number of paths between pairs of nodes traversing e. Now, suppose that the operator chooses collection T to be used for communications and the adversary targets link e. As a result of the adversary's attack, link e is removed from the graph. Consequently, the traffic (or the paths) that were traversing link e are lost[5], and the network operator sustains a loss of $\lambda(T, e)$. Therefore, the function $\lambda(T, e)$ will be called the usage or loss function of the communication model.

Before introducing the formal definition of the game, I summarize a previously proposed communication model to illustrate the concepts discussed so far.

Supply-Demand Communication Model

The Supply-Demand (S-D) communication model was introduced by Gueye and Marbukh in [Gueye and Marbukh, 2012] for scenarios where the operator's goal is to carry a fixed amount of goods from a nonempty set of "source" nodes to a nonempty set of "destination" nodes using the links of the network. This formulation can be used to model various problems related to the attack-resilient transportation of goods, such as electric power transmission, water supply, and other critical infrastructure facilities.

In the S-D communication model, the network topology is assumed to be a directed graph G. For each node $v \in \mathcal{V}$, there is both a supply $s(v) \in \mathbb{N}$ and a demand $d(v) \in \mathbb{N}$ value given. The links are assumed to be uncapacitated, that is, each link can carry an unlimited amount of goods.[6] Finally, the links are assumed to be able to carry only integer amounts of goods.

To transport the goods, the network operator chooses a collection of links forming a feasible (integer) flow. A feasible flow $T : \mathcal{E} \to \mathbb{N}$ is a function which assigns, to each link e, the amount of goods $T(e)$ it carries, satisfying the conservation of flow constraint at each node. More specifically, for each node $v \in \mathcal{V}$, the sum of the outgoing flows $\sum_{(v,u)\in\mathcal{E}} T((v, u))$ minus the sum of the incoming flows $\sum_{(u,v)\in\mathcal{E}} T((u, v))$ has to be $s(v) - d(v)$. To attack the network, the adversary targets a link $e \in \mathcal{E}$. Hence, the players' pure-strategy sets are the set of all feasible flows $\mathcal{T}$ (i.e., the set of feasible collections) and $\mathcal{E}$. Finally, the usage (or loss) function is defined in the most natural way: $\lambda(T, e)$ is simply the amount of goods $T(e)$ that flow T assigns to link e. In other words, if a links is removed, then the goods transported on that link are considered lost.

2.2.1 Game-Theoretic Measure of Robustness

Given a network topology (that is, a graph G) and a communication model (that is, a set of feasible collections $\mathcal{T}$ and a usage function λ), a two-player game is defined between the network operator and a strategic adversary as follows.

The network operator's goal is to provide some form of connectivity between the nodes by choosing a feasible collection from the set $\mathcal{T}$ to be used for communications. Hence, the operator's pure-strategy set is the set of feasible collections $\mathcal{T}$. Meanwhile, the adversary's goal is to disrupt the communication between the nodes by choosing – simultaneously with the operator – a link from the set $\mathcal{E}$ to be attacked (i.e., to be removed from the network). Hence, the adversary's pure-strategy set is the set of links $\mathcal{E}$. To carry out a successful attack against link $e \in \mathcal{E}$, the adversary has to spend some effort, which is denoted by μ_e. Since these attack costs could be prohibitively high, making every attack against a given operator strategy economically infeasible for the adversary, the adversary also has the option of not attacking. Formally, the attacker's pure-strategy set is complemented with a "no-attack" strategy, and whenever the attacker chooses this strategy, both players receive zero payoff (or loss). Since this no-attack option can easily be modeled by adding a loop with zero attack cost to the graph, I will disregard it in the subsequent sections.

The players' pure-strategy payoffs are defined as follows: when the operator selects collection T and the attacker targets link e, the operator sustains a loss of $\lambda(T, e)$ (as defined above), and the attacker receives a net reward of $\lambda(T, e) - \mu_e$. We consider mixed-strategy Nash equilibria, where the network

[5] Or – in a more general interpretation – the traffic traversing link e is compromised.

[6] Since this communication model is used only as an illustration, I restrict its discussion to the special case of uncapacitated networks. For a discussion of the more general, capacitated network model, I refer the interested reader to [Gueye and Marbukh, 2012].

operator chooses a distribution $\boldsymbol{\alpha}$ over the set $\mathcal{T}$, and the attacker chooses a distribution $\boldsymbol{\beta}$ over the set $\mathcal{E}$. We assume that the operator tries to minimize her expected loss, while the attacker tries to maximize her expected net reward. Formally, the operator chooses $\boldsymbol{\alpha}$ to minimize her expected loss

$$\sum_{T\in\mathcal{T}}\sum_{e\in\mathcal{E}}\alpha_T\beta_e\lambda(T,e)\ , \tag{2.1}$$

while the attacker chooses $\boldsymbol{\beta}$ to maximize her expected net reward

$$\sum_{T\in\mathcal{T}}\sum_{e\in\mathcal{E}}\alpha_T\beta_e\left(\lambda(T,e)-\mu_e\right)\ . \tag{2.2}$$

Finally, let θ_{max} be the attacker's expected equilibrium payoff, which – as we will see in the following subsection – has been shown to be the same in every equilibrium [Gueye, 2011]. Since it is the same in all the equilibria, the value of θ_{max} is uniquely defined in a game.

Now, consider the value of θ_{max} for a given network. If the network is vulnerable, that is, if the network is easy to attack, then the adversary can cause serious damage at little expense, and θ_{max} has to be high. If – on the other hand – the network is robust, the adversary has to spend a lot of effort to cause some damage, and θ_{max} has to be low. Thus, we can use θ_{max} to quantify the vulnerability of a network and use its inverse to quantify robustness. Formally, we let the game-theoretic vulnerability (or robustness) of a graph G be denoted by $\theta_{max}(G)$ (or $\theta^{-1}_{max}(G)$). The following subsection gives a characterization of θ_{max} using the theory of blocking pairs of polyhedra.

2.2.2 Solving the Game

In [Gueye, 2011], the Nash equilibria of blocking games are characterized using the theory of blocking pairs of polyhedra (BPP). Here, I introduce the basic concepts of BPP that are essential for understanding this characterization, and refer the interested reader to [Fulkerson, 1971] for a more detailed discussion.

The *polyhedron* $P_{\boldsymbol{\Lambda}}$ of a nonnegative $N\times m$ matrix $\boldsymbol{\Lambda}$ is defined as the vector sum of the convex hull of the rows $\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_N$ of $\boldsymbol{\Lambda}$ and the nonnegative orthant; formally, $P_{\boldsymbol{\Lambda}}=\text{conv.hull}(\boldsymbol{\lambda}_1,\ldots,\boldsymbol{\lambda}_N)+\mathbb{R}^m_{\geq 0}$. In other words, the polyhedron $P_{\boldsymbol{\Lambda}}$ consists of vectors which can be expressed as a sum of a non-negative vector and a convex linear combination of the rows of $\boldsymbol{\Lambda}$. The *blocker* $bl(P_{\boldsymbol{\Lambda}})$ of $P_{\boldsymbol{\Lambda}}$ is defined as

$$bl(P_{\boldsymbol{\Lambda}})=\left\{\boldsymbol{y}\in\mathbb{R}^m_{\geq 0}\ \middle|\ \forall\boldsymbol{x}\in P_{\boldsymbol{\Lambda}}:\ \boldsymbol{x}'\boldsymbol{y}\geq 1\right\}\ . \tag{2.3}$$

Alternatively, the blocker $bl(P_{\boldsymbol{\Lambda}})$ can also be defined as the set of vectors that "block" every row of $\boldsymbol{\Lambda}$; formally, $bl(P_{\boldsymbol{\Lambda}})=\{\boldsymbol{y}\in\mathbb{R}^m_{\geq 0}\ :\ \boldsymbol{\Lambda y}\geq\mathbf{1}\}$. Note that the blocker of a polyhedron itself is also a polyhedron.

Now, let $\boldsymbol{\Lambda}$ be the loss (i.e., negative payoff) matrix of the defender; formally, $\Lambda_{S,e}=\lambda(S,e)$. Using the notation introduced above, $P_{\boldsymbol{\Lambda}}$ is the polyhedron associated with $\boldsymbol{\Lambda}$, and $bl(P_{\boldsymbol{\Lambda}})$ is the blocker of $P_{\boldsymbol{\Lambda}}$. Before characterizing the equilibria of a blocking game using its blocker $bl(P_{\boldsymbol{\Lambda}})$, I have to introduce a few more notions. First, let $\Omega=\{\boldsymbol{\omega}_1,\ldots,\boldsymbol{\omega}_K\}$ be the set of the extreme points of the blocker $bl(P_{\boldsymbol{\Lambda}})$ (intuitively, the "vertices" of the blocker). Next, for a vector $\boldsymbol{y}\in bl(P_{\boldsymbol{\Lambda}})$, let the quantity $\theta(\boldsymbol{y})$ be defined as

$$\theta(\boldsymbol{y})=\frac{1}{\boldsymbol{y'1}}\left(1-\boldsymbol{y}'\boldsymbol{\mu}\right)\ , \tag{2.4}$$

and let $\theta_{max}=\max_{\boldsymbol{y}\in bl(P_{\boldsymbol{\Lambda}})}\theta(\boldsymbol{y})$. Note that we will soon see that this is not an abuse of notation, as the maximum of $\theta(\boldsymbol{y})$ is indeed the adversary's equilibrium payoff. In [Gueye, 2011], it was shown that the maximum θ_{max} is attained at an extreme point $\boldsymbol{\omega}$ (or at some extreme points) of the blocker; that is, $\max_{\boldsymbol{y}\in bl(P_{\boldsymbol{\Lambda}})}\theta(\boldsymbol{y})=\max_{\boldsymbol{\omega}\in\Omega}\theta(\boldsymbol{\omega})$. Finally, let Ω_{max} denote the set of extreme points for which the maximum is attained; formally, let $\Omega_{max}=\{\boldsymbol{\omega}\in\Omega\mid\theta(\boldsymbol{\omega})=\theta_{max}\}$.

Theorem 1 (Gueye [Gueye, 2011])**.** *In any blocking game, the following always hold.*

1. *If $\theta_{max}\leq 0$, then not attacking is always optimal for the adversary.*
2. *If $\theta_{max}\geq 0$, then for every probability distribution $\boldsymbol{\gamma}$ over Ω_{max}, the adversary's strategy $\boldsymbol{\beta}$ defined by*

$$\beta_e=\sum_{\boldsymbol{\omega}\in\Omega_{max}}\gamma_{\boldsymbol{\omega}}\frac{\omega_e}{\boldsymbol{\omega'1}} \tag{2.5}$$

is in Nash equilibrium with any strategy $\boldsymbol{\alpha}$ of the defender that satisfies the following properties:

$$\begin{cases} \sum_{S\in\mathcal{S}} \alpha_S \lambda(S,e) - \mu_e = \theta_{max}, & \forall e \in \mathcal{E} \text{ s. t. } \beta_e > 0 \ , \\ \sum_{S\in\mathcal{S}} \alpha_S \lambda(S,e) - \mu_e \leq \theta_{max}, & \forall e \in \mathcal{E} \ . \end{cases}$$

Furthermore, there exists at least one such strategy $\boldsymbol{\alpha}$. The corresponding payoffs are θ_{max} for the adversary and $\sum_{\boldsymbol{\omega}\in\Omega_{max}} \frac{\gamma_{\boldsymbol{\omega}}}{\boldsymbol{\omega}'\mathbf{1}}$ for the defender.

3. *If $\boldsymbol{\mu} = \mathbf{0}$, then every Nash equilibrium pair of strategies is of the above type.*

For the proof of the theorem, see [Gueye, 2011].

2.3 Computational Complexity of Solving Network Blocking Games in General

In this section, I discuss the computational complexity of solving network blocking games in general. First, note that computing a Nash equilibrium in a general two-player game is a PPAD-complete[7] problem [Chen and Deng, 2006]. Zero-sum two-player games, on the other hand, can be cast as linear programs and – consequently – can be solved in polynomial time using linear programming tools. However, these results assume that the input of the computational problem is the payoff matrix of the game, while in network blocking games, the input is actually the network topology whose robustness we have to quantify. In other words, the payoff matrix of a network blocking game is only implicitly given. Moreover, in all of the proposed communication models, the size of the payoff matrix is generally exponential in the size of the network due to the cardinality of the operator's strategy set. Therefore, conventional algorithms, which solve games using their payoff matrices, cannot be used to solve network blocking games efficiently, since computing the input of these algorithms (i.e., the payoff matrix) is not possible in polynomial time.

I discuss the computational complexity of solving network blocking games as follows. First, in Section 2.3.1, I show that solving these game is indeed generally hard, by proving that the problem of finding the adversary's equilibrium payoff is NP-hard. Then, in Section 2.3.2, I focus on a particularly interesting subclass of network blocking games, whose polyhedra can be characterized using a polynomial number of linear inequalities, and show that these games can be solved efficiently. Note that, since my primary interest lies in quantifying the robustness of network topologies, I will primarily focus on the complexity of computing the adversary's equilibrium payoff. Nevertheless, I will also discuss the complexity of finding an equilibrium strategy profile.

2.3.1 NP-Hardness

In this subsection, I show that solving a network blocking game is NP-hard in general. Note that, if we put no constraints on the usage (or loss) function λ, the problem would obviously be hard, since one could simply choose λ to be function that is hard to compute. For example, one could let the value of $\lambda(T,e)$ be the solution to a Knapsack Problem whose items are the elements of T. Therefore, I will assume that computing the value of the usage function λ and testing whether a collection T is feasible are both possible in polynomial time.

I prove the hardness of solving network blocking games by reducing a well-known NP-hard problem, the *Knapsack Problem* (KP), to the problem of computing the attacker's equilibrium payoff, which I formalize as the *Equilibrium Problem* (EP). The decision versions of KP and EP are formally defined as follows.

Definition 1 (Knapsack Problem [KP])**.** Given N items, where item i has weight c_i and value v_i, a capacity C, and a value V, is there a subset S whose sum weight is at most C, i.e., $\sum_{i\in S} c_i \leq C$, and whose sum value is at least V, i.e., $\sum_{i\in S} v_i \geq V$?

[7] PPAD (Polynomial Parity Arguments on Directed graphs) is a class of problems that are believed to be hard, lying "between" P and NP.

Definition 2 (Equilibrium Problem [EP])**.** Given a set of elements $\mathcal{E}$, a polynomial-time function $I_{T\in\mathcal{T}}$ for testing $T \in \mathcal{T}$, a polynomial-time usage function $\lambda(T, e)$, a vector of attack costs $\boldsymbol{\mu} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$, and a payoff value p, is the adversary's equilibrium payoff less than or equal to p?

Note that, instead of solving the game "completely" (i.e., finding an equilibrium strategy profile and the adversary's equilibrium payoff), the Equilibrium Problem requires only comparing the adversary's equilibrium payoff with a threshold value, which is a much easier problem to solve. The easiness of EP allows us to prove the hardness of all problems related to solving the game through reduction. More specifically, if we had a polynomial-time algorithm for computing an equilibrium strategy profile, we would also have an algorithm for computing the adversary's equilibrium payoff and – consequently – for solving EP. Thus, if the EP is NP-hard, so is solving the game.

The following theorem shows that the Equilibrium Problem is NP-hard.

Theorem 2. *The Knapsack Problem is polynomial-time reducible to the Equilibrium Problem.*

Proof. Given an instance $(\boldsymbol{c}, \boldsymbol{v}, C, V)$ of the Knapsack Problem, we construct an instance $(\mathcal{E}, I_{T\in\mathcal{T}}, \lambda(T, e), p)$ of the Equilibrium Problem as follows.

- Let the set of elements be $\mathcal{E} = \{1, \ldots, N\}$,
- let the feasibility testing function be $I_{T\in\mathcal{T}} = \begin{cases} \text{true} & \text{if } \sum_{i\in T} c_i \leq C \\ \text{false} & \text{otherwise,} \end{cases}$ [8]
- let the usage function be $\lambda(T, e) = \frac{1}{\sum_{i\in T} v_i}$,
- let the adversary's attack costs be $\boldsymbol{\mu} = \mathbf{0}$,
- and let the threshold payoff value be $p = \frac{1}{V}$.

It is easy to see that both $I_{T\in\mathcal{T}}$ and $\lambda(T)$ can be computed in polynomial time, as they only require computing the sum of a given set and then comparing it with a constant or calculating its reciprocal. Furthermore, every step of the reduction can also be carried out in time and space that is polynomial in the size of the Knapsack Problem instance. Hence, the reduction itself can be done in polynomial time.

I claim that the given instance of KP is true if and only if the above instance of EP is true. To prove this, we have to show that there exists a subset $S \subseteq \{1, \ldots, N\}$ whose sum weight is at most W and whose sum value is at least V *if and only if* the adversary's equilibrium payoff in the above game is less than or equal to p.

First, assume that there exists a subset S satisfying the constraints of the Knapsack Problem. Then, we have to show that the adversary's equilibrium payoff is at most p. Let $\boldsymbol{\alpha}^*$ be the mixed strategy that uses only subset S. Formally, let $\alpha_S^* = 1$ and, for every other subset $U \neq S$, let $\alpha_U^* = 0$. If the operator uses this strategy, then

$$\forall e: \ \lambda(S, e) = \frac{1}{\sum_{i\in S} v_i} = \frac{1}{V} = p \ , \tag{2.6}$$

which implies that her expected loss is p regardless of the strategy of the adversary. Therefore, the operator's equilibrium loss and, hence, the adversary's equilibrium payoff have to be at most the threshold value p.

Second, assume that there does not exist a subset satisfying the constraints of the Knapsack Problem. In this case, we have to show that the adversary's equilibrium payoff is greater than p. Since no subset satisfies the constraints of KP, we have that $\sum_{i\in T} v_i < V$ for every $T \in \mathcal{T}$. Consequently,

$$\forall T, e: \ \lambda(T, e) = \frac{1}{\sum_{i\in T} v_i} > \frac{1}{V} = p \ . \tag{2.7}$$

Consequently, the operator's expected loss for any strategy profile $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is

$$\sum_{T\in\mathcal{T}} \alpha_T \sum_{e\in\mathcal{E}} \beta_e \underbrace{\lambda(T, e)}_{>p} > p \ . \tag{2.8}$$

Therefore, the adversary's equilibrium payoff has to be greater than the threshold value p. □

[8] That is, let $\mathcal{T} = \{T \subseteq \{1, \ldots, N\} \mid \sum_{i\in T} c_i \leq C\}$.

2.3.2 Linear Programming Solution

While solving network blocking games is NP-hard in general, there exists a number of communication models for which the adversary's equilibrium payoff (and, for some models, even an equilibrium strategy profile) can be computed efficiently. In this subsection, I discuss the computational complexity of models for which the polyhedron $P_{\boldsymbol{\Lambda}}$ has a polynomial-size linear characterization[9] and show that these models can be solved efficiently. First, I derive a polynomial-size characterization of the blocker $bl(P_{\boldsymbol{\Lambda}})$ based on the characterization of the polyhedron $P_{\boldsymbol{\Lambda}}$ (Lemma 1). Then, I show how to compute the adversary's equilibrium payoff θ_{max} using a linear programming solver and the characterization of the blocker. Finally, I discuss finding an equilibrium strategy profile.

Assume that the polyhedron $P_{\boldsymbol{\Lambda}}$ of the game has a polynomial-size linear characterization

$$P_{\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \wedge \boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \right) \right\} , \tag{2.9}$$

where $\boldsymbol{f}$ is a vector variable of polynomial length (i.e., k is a polynomial function of the network size), while $\boldsymbol{S} \in \mathbb{R}^{|\mathcal{E}| \times k}_{\geq 0}$, $\boldsymbol{C} \in \mathbb{R}^{l \times k}_{\geq 0}$, and $\boldsymbol{c} \in \mathbb{R}^{l}_{\geq 0}$ are constants of polynomial size. Intuitively, one can think of $\mathbb{R}^{k}_{\geq 0}$ as a search space, whose dimension is polynomial in the size of the network, $\boldsymbol{S}$ as a mapping from the search space to the vector space of the polyhedron, and the inequality based on $\boldsymbol{C}$ and $\boldsymbol{c}$ as a constraint on the search space.

Based on the above characterization, the following lemma gives a polynomial-size characterization of the blocker.

Lemma 1. *The blocker of the polyhedron defined as*

$$P_{\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \mid \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \wedge \boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \right) \right\} , \tag{2.10}$$

where $\boldsymbol{S} \in \mathbb{R}^{|\mathcal{E}| \times k}_{\geq 0}$, $\boldsymbol{C} \in \mathbb{R}^{l \times k}_{\geq 0}$, *and* $\boldsymbol{c} \in \mathbb{R}^{l}_{\geq 0}$, *can be characterized as*

$$bl(P_{\boldsymbol{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \mid \exists \boldsymbol{g} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}, \boldsymbol{h} \in \mathbb{R}^{l}_{\geq 0} \left(\boldsymbol{g} \leq \boldsymbol{y} \wedge \boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{S}'\boldsymbol{g} \wedge \boldsymbol{c}'\boldsymbol{h} \geq 1 \right) \right\} . \tag{2.11}$$

Proof. I prove Equation (2.11) in two steps:

- Right-hand side (RHS) of Equation (2.11) $\subseteq bl(P_{\boldsymbol{\Lambda}})$: We have to show that every element of the RHS of (2.11) is an element of the blocker $bl(P_{\boldsymbol{\Lambda}})$. Let $\tilde{\boldsymbol{y}}$ be an arbitrary element of the RHS, that is, a vector which satisfies the constraints of the RHS with some $\tilde{\boldsymbol{g}}$ and $\tilde{\boldsymbol{h}}$. To prove that $\tilde{\boldsymbol{y}} \in bl(P_{\boldsymbol{\Lambda}})$, we have to show that $\tilde{\boldsymbol{y}}'\boldsymbol{x} \geq 1$ for every $\boldsymbol{x} \in P_{\boldsymbol{\Lambda}}$. We can formulate this as a linear programming problem as follows:

  $$\text{Minimize } \tilde{\boldsymbol{y}}'\boldsymbol{x} \tag{2.12}$$

  subject to

  $$\boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \tag{2.13}$$
  $$\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} , \tag{2.14}$$

  where $\boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0}$ and $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$.

  Observe that the constraints of the linear program correspond to the characterization of $P_{\boldsymbol{\Lambda}}$. Consequently, the above linear program's set of feasible solutions projected to $\boldsymbol{x}$ is actually $P_{\boldsymbol{\Lambda}}$. Therefore, it suffices to show that the value of the linear program is at least 1. To see this, consider the dual linear program:

  $$\text{Maximize } \boldsymbol{c}'\boldsymbol{h} \tag{2.15}$$

  subject to

  $$\boldsymbol{g} \leq \tilde{\boldsymbol{y}} \tag{2.16}$$
  $$\boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{S}'\boldsymbol{g} , \tag{2.17}$$

[9] I say that a polyhedron has a *polynomial-size linear characterization* if it can described using Equation (2.9).

where $\boldsymbol{g} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$, and $\boldsymbol{h} \in \mathbb{R}^{l}_{\geq 0}$.

Since $\tilde{\boldsymbol{y}}$ satisfies the constraints of the RHS of Equation (2.11) with $\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{h}}$, we have that $(\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{h}})$ is a feasible solution. Furthermore, we also have that the objective function for this solution is at least 1, since $\boldsymbol{c}'\tilde{\boldsymbol{h}} \geq 1$. Thus, the value of the dual program has to be at least 1. From linear programming duality, it follows readily that the value of the primal program is also greater than or equal to 1, which proves that $\tilde{\boldsymbol{y}}$ blocks every element of the polyhedron P_Λ.

- $bl(P_{\boldsymbol{\Lambda}}) \subseteq$ RHS of Equation (2.11): We have to show that every $\tilde{\boldsymbol{y}} \in bl(P_{\boldsymbol{\Lambda}})$ satisfies the constraints of the RHS. To see this, first consider the linear program from the first part of the proof. Since $\tilde{\boldsymbol{y}}$ blocks every $\boldsymbol{x} \in P_{\boldsymbol{\Lambda}}$, we have that the value of the linear program (and its dual) is at least 1. Now, consider an optimal solution $(\tilde{\boldsymbol{g}}, \tilde{\boldsymbol{h}})$ of the dual linear program. Since the value of the dual linear program is at least 1, we have that $1 \leq \boldsymbol{c}'\tilde{\boldsymbol{h}}$. Furthermore, we also have $\tilde{\boldsymbol{g}} \leq \tilde{\boldsymbol{y}}$ and $\boldsymbol{C}'\tilde{\boldsymbol{h}} \leq \boldsymbol{S}'\tilde{\boldsymbol{g}}$ from the constraints of the linear program. Thus, $\tilde{\boldsymbol{y}}$ satisfies the constraints of the RHS of Equation (2.11) with $\tilde{g}$ and $\tilde{\boldsymbol{h}}$.

□

Recall that our primary goal is to compute $\theta_{max} = \max_{\boldsymbol{y} \in bl(P_{\boldsymbol{\Lambda}})} \theta(\boldsymbol{y})$ in polynomial time. The most straightforward solution is to formulate this as an optimization problem subject to the set of linear constraints given by the above polynomial-size characterization. Unfortunately, the objective function θ cannot be expressed as a linear function of $\boldsymbol{y}$ because of the division with $\mathbf{1}'\boldsymbol{y}$. Thus, to formulate the problem as a linear program, we introduce a variable ϕ, which measures $\frac{1}{\mathbf{1}'\boldsymbol{y}}$, and scale the original variables. The resulting linear program is

$$\text{Maximize } \phi - \boldsymbol{\mu}'\boldsymbol{\beta} \tag{2.18}$$

subject to

$$\mathbf{1}'\boldsymbol{\beta} = 1 \tag{2.19}$$
$$\boldsymbol{g} \leq \boldsymbol{\beta} \tag{2.20}$$
$$\boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{S}'\boldsymbol{g} \tag{2.21}$$
$$\boldsymbol{c}'\boldsymbol{h} \geq \phi\ , \tag{2.22}$$

where $\phi \in \mathbb{R}_{\geq 0}$, $\boldsymbol{\beta}, \boldsymbol{g} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$, and $\boldsymbol{h} \in \mathbb{R}^{l}_{\geq 0}$. First, observe that the objective function is indeed $\phi - \boldsymbol{\mu}'\boldsymbol{\beta} = \frac{1}{\mathbf{1}'\boldsymbol{y}} - \boldsymbol{\mu}'\left(\frac{1}{\mathbf{1}'\boldsymbol{y}}\boldsymbol{y}\right) = \frac{1}{\mathbf{1}'\boldsymbol{y}}(1 - \boldsymbol{\mu}'\boldsymbol{y}) = \theta(\boldsymbol{y})$. Second, notice that the constant 1 is replaced by ϕ in the last constraint due to the scaling. Finally, notice that we have to introduce a new constraint to ensure that $\mathbf{1}'\boldsymbol{\beta} = \mathbf{1}'\frac{\boldsymbol{y}}{\mathbf{1}'\boldsymbol{y}} = 1$ holds. As the size of the above linear program is polynomial in the size of the network, we can compute θ_{max} efficiently using any standard linear programming solver.

Finding an Equilibrium Strategy Profile

Now, I discuss the problem of finding an equilibrium strategy profile in models for which the polyhedron $P_{\boldsymbol{\Lambda}}$ has a polynomial-size linear characterization. First, I show how to find an adversarial equilibrium strategy, that is, how to find a mixed adversarial strategy $\boldsymbol{\beta}$ for which there exists at least one mixed operator strategy $\boldsymbol{\alpha}$ such that $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ is an equilibrium strategy profile. Second, I discuss the more complicated problem of finding an equilibrium operator strategy.

If θ_{max} is less than or equal to zero, then not attacking is an equilibrium strategy for the adversary. Otherwise, an optimal solution $\boldsymbol{\beta}^*$ to the above scaled linear program is an equilibrium adversarial strategy. More specifically, I claim that, if the adversary uses $\boldsymbol{\beta}^*$ as her mixed strategy, then her expected payoff is at least θ_{max} regardless of the strategy of the operator. To see this, consider the element of the blocker $\boldsymbol{y} = \frac{\boldsymbol{\beta}^*}{\phi^*}$, where ϕ^* is from the same optimal solution as $\boldsymbol{\beta}^*$. Since $\boldsymbol{y} \in bl(P_{\boldsymbol{\Lambda}})$, we have $\boldsymbol{\alpha}\boldsymbol{\Lambda}\boldsymbol{y} \geq 1$ for every mixed operator strategy $\boldsymbol{\alpha}$ (i.e., for every distribution over the feasible collections). Then, for

any mixed operator strategy $\boldsymbol{\alpha}$, the adversary's expected payoff is

$$\begin{aligned}
&\boldsymbol{\alpha}\boldsymbol{\Lambda}\boldsymbol{\beta}^* - \boldsymbol{\mu}'\boldsymbol{\beta}^* && (2.23)\\
=&\boldsymbol{\alpha}\boldsymbol{\Lambda}(\phi^*\boldsymbol{y}) - \boldsymbol{\mu}'\boldsymbol{\beta}^* && (2.24)\\
=&\phi^* \underbrace{\boldsymbol{\alpha}\boldsymbol{\Lambda}\boldsymbol{y}}\geq 1 - \boldsymbol{\mu}'\boldsymbol{\beta}^* && (2.25)\\
\geq&\phi^* - \boldsymbol{\mu}'\boldsymbol{\beta}^* \ , && (2.26)
\end{aligned}$$

which is greater than or equal to θ_{max}. Thus, with the above scaled linear program, we can also compute an adversarial equilibrium strategy efficiently besides computing the adversary's equilibrium payoff.

Computing an equilibrium strategy profile – that is, computing not only the adversary's strategy but the operator's strategy as well – is more complicated. First of all, recall that the canonical representation of a mixed strategy for the operator is a distribution over the set of all feasible collections $\mathcal{T}$, whose cardinality can be exponential in the size of the network. Thus, any algorithm that outputs a mixed operator strategy in its canonical representation has exponential running time in general. Consequently, computing an equilibrium strategy profile efficiently is possible only if the cardinality of the support of the operator's mixed strategy is polynomial in the size of the network, and we output only these non-zero coefficients.

From Theorem 1, we have that the operator's equilibrium strategy $\boldsymbol{\alpha}$ has to satisfy

$$\forall e \in \mathcal{E} \text{ such that } \beta_e > 0 : \ \sum_{T\in\mathcal{T}} \alpha_T \lambda(T,e) - \mu_e = \theta_{max} \tag{2.27}$$

and

$$\forall e \in \mathcal{E} : \ \sum_{T\in\mathcal{T}} \alpha_T \lambda(T,e) - \mu_e \leq \theta_{max} \ . \tag{2.28}$$

Again, we cannot solve this efficiently directly due to the size of $\boldsymbol{\alpha}$. However, the constraints are actually not directly on $\boldsymbol{\alpha}$ but rather on the corresponding element $\boldsymbol{x} = \boldsymbol{\alpha}\boldsymbol{\Lambda}$ of the polyhedron $P_{\boldsymbol{\Lambda}}$. Therefore, we can first find an element $\boldsymbol{x}$ that satisfies

$$\forall e \in \mathcal{E} \text{ such that } \beta_e > 0 : \ x_e - \mu_e = \theta_{max} \tag{2.29}$$

and

$$\forall e \in \mathcal{E} : \ x_e - \mu_e \leq \theta_{max} \ , \tag{2.30}$$

and then search for a corresponding $\boldsymbol{\alpha}$. The first step can be performed efficiently using these linear constraints, the polynomial-size linear characterization from Equation 2.9, and some linear programming solver.

Unfortunately, there is no efficient and general algorithm for computing a mixed operator $\boldsymbol{\alpha}$ strategy from an element $\boldsymbol{x}$ of the polyhedron $P_{\boldsymbol{\Lambda}}$. Solving $\boldsymbol{\alpha}\boldsymbol{\Lambda} \leq \boldsymbol{x}$ subject to $\boldsymbol{\alpha}$ being a distribution is possible using a linear programming solver, but its running time is exponential due to the sizes of $\boldsymbol{\alpha}$ and $\boldsymbol{\Lambda}$. However, for the communication models proposed in Sections 2.4 and 2.5, I will provide polynomial-time algorithms for finding a mixed operator strategy given an element of the polyhedron $P_{\boldsymbol{\Lambda}}$.

Example: Polyhedron and Blocker of the Supply-Demand Communication Model

As an example, consider the Supply-Demand communication model, whose polyhedron $P_{\boldsymbol{\Lambda}}$ can be characterized as

$$P_{\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|} \;\middle|\; \exists f : \mathcal{E} \to \mathbb{R}_{\geq 0} \Big(\quad \forall e \in \mathcal{E} : \ f(e) \leq x_e \right.$$
$$\left. \wedge \ \forall v \in \mathcal{V} : \sum_{(u,v)\in\mathcal{E}} f((u,v)) - \sum_{(v,u)\in\mathcal{E}} f((v,u)) = s(v) - d(v) \Big) \right\} , \tag{2.31}$$

where $s(v)$ and $d(v)$ are the supply and demand at node v, respectively [Laszka and Gueye, 2013b]. Using Lemma 1, we can express the blocker $bl(P_{\mathbf{\Lambda}})$ as

$$bl(P_{\mathbf{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|} \;\middle|\; \exists \pi : \mathcal{V} \to \mathbb{R} \Big(\quad \forall e = (u,v) \in \mathcal{E} : \; \pi(u) - \pi(v) \leq y_e \right.$$
$$\left. \wedge \sum_{v \in \mathcal{V}} \pi(v) \left(s(v) - d(v) \right) \geq 1 \Big) \right\} . \tag{2.32}$$

2.4 All-to-One Communications Model

Many access and sensor networks are inherently vulnerable to physical attacks, such as the jamming of wireless signals or the destruction of nodes and links. From a topological point of view, the common characteristic of these networks is that the primary goal of the nodes is to communicate with a (set of) designated nodes; hence, I will refer to them as all-to-one networks. For example, in a mesh network of wireless routers that provides Internet access to mobile terminals, every router is typically interested in communicating with a designated gateway router, through which the Internet is reachable, and not with other peer routers of the network (except for the purpose of packet forwarding of course). As another example, in a sensor network, the goal of the network is to collect the sensed data at a designated central node. In this section, I introduce a communications model for these networks, which I call the All-to-One communication model.

In the All-to-One communications model, the goal of the network operator is to enable all nodes to communicate with a designated node $r \in \mathcal{V}$. Note that, in practice, there can be multiple designated nodes; however, for the sake of readability, I restrict the analysis to a single designated node for now. In Section 2.4.2, I will show how multiple designated nodes can be modeled using a single "super"-designated node. The network topology is assumed to be given in the form of a directed graph G.[10] To connect the nodes to the designated node r, the network operator chooses a collection of links T that forms a spanning reverse arborescence rooted at r[11]. (Note that – again, for readability – I will often shorten the term "spanning reverse arborescence rooted at r" to simply "reverse arborescence". Nevertheless, all arborescences mentioned in this section are spanning reverse arborescences rooted at r.) Hence, the set of feasible collections $\mathcal{T}$ is the set of all reverse arborescences. In practice, this reverse arborescence can be implemented as, for example, the next-hop forwarding table entries for r, which are stored at the individual nodes of the network.

Now, I define the usage (or loss) function $\lambda(T, e)$. Let the network be connected using reverse arborescence T. Then, the amount of traffic on link $e \in \mathcal{E}$ is proportional to the number of nodes that use link e to communicate with the designated node r. Hence, the usage of link e in arborescence T is the number of nodes connected to r by paths including e.[12] Alternatively, we can motivate the same definition by the amount of loss sustained. If link $e \in E$ is attacked, then those nodes which were connected to r by paths including e can no longer communicate with r, and can be considered lost for the network operator. Thus, for a given reverse arborescence T and link e, the usage $\lambda(T, e)$ is the *number of those nodes that are disconnected from r in $G[T \setminus \{e\}]$*.

2.4.1 Solving the Game

In this subsection, I provide a polynomial-size linear characterization for the polyhedron $P_{\mathbf{\Lambda}}$ associated with the All-to-One communication model. Recall from Section 2.3.2 that such a characterization can be used to solve the game efficiently. Furthermore, I begin with a lemma (Lemma 2) whose constructive proof can be used to compute a mixed operator strategy from an arbitrary element of the polyhedron $P_{\mathbf{\Lambda}}$; hence, it can be used for finding an equilibrium strategy profile.

[10] In practice, links can also be bidirectional, especially in the case of wireless networks. However, since the results and corresponding proofs for undirected graphs (and spanning trees instead of reverse arborescences) are almost identical to those for directed graphs, I omit them.

[11] A directed, rooted spanning tree in which all edges point to the root r.

[12] Since T is an arborescence (i.e., directed tree), there is a unique path between any pair of nodes. Hence, the number of path to r that include e is well-defined.

Lemma 2. *Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with designated node $r \in \mathcal{V}$, and let $\mathbf{\Lambda}$ be the usage (or loss) matrix of the All-to-One communication model. Then, for any $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$, $\boldsymbol{x} \in P_{\mathbf{\Lambda}}$ if there exists a feasible uncapacitated multi-source flow f in G such that $\forall e \in \mathcal{E} : f(e) \leq x_e$ and every $v \in \mathcal{V} \setminus \{r\}$ is a source producing an amount of 1, while r is a sink consuming an amount of $|\mathcal{V}| - 1$.*

Proof. We have to show that the lemma holds for every $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$. Let $\boldsymbol{x} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$ be an arbitrary vector for which a feasible uncapacitated multi-source flow f satisfying $\forall e \in \mathcal{E} : f(e) \leq x_e$ exists. We have to show that there is a distribution $\boldsymbol{\alpha}$ over the set of reverse arborescences such that $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq x_e$ holds for every edge $e \in \mathcal{E}$. I prove this by showing that there exists a distribution $\boldsymbol{\alpha}$ such that $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq f(e)$ for every $e \in \mathcal{E}$. In other words, I show that any feasible uncapacitated multi-source flow f can be expressed as a convex combination of weighted reverse arborescences. The proof is constructive and it is based on the following algorithm.

1. Eliminate all cyclic flows from f.
2. Find a spanning reverse arborescence T rooted at r consisting of only edges e with $f(e) > 0$.
3. Let $\alpha_T := \min_{e \in T} \frac{f(e)}{\lambda(T,e)}$.
4. For every $e \in \mathcal{E}$, let $f(e) := f(e) - \alpha_T \cdot \lambda(T, e)$.
5. If the amount of flow transported by f is greater than zero, then continue from Step 2.
6. Let $\alpha_T := 0$ for every other reverse arborescence.

Before proving the correctness of the algorithm, we first have to show that Step 2 can be executed in each iteration, otherwise the algorithm would terminate erroneously. If f is a multi-source network flow in which each non-designated node is a source producing a non-zero amount, there has to be a directed path from every non-designated node to r consisting of only edges with positive flow amounts. Hence, if f is a multi-source flow, a reverse arborescence has to exist. Thus, it suffices to show that, if f is a network flow transporting γ from every non-designated to r before Step 4, then it is a network flow transporting $\gamma - \alpha_T$ from every non-designated node to r after Step 4.

For a given node $v \in \mathcal{V} \setminus \{r\}$, let λ_v denote $\lambda(T, e_{out})$, where e_{out} is the outgoing edge of v in T. It is easy to see that the sum of $\lambda(T, e_{in})$ over all incoming edges $e_{in} \in \mathcal{E}$ of v is $\lambda_v - 1$. Since the flow along every edge e is decreased by $\alpha_T \lambda(T, e)$, the sum of the outgoing flows is decreased by $\alpha_T \lambda_v$. Similarly, the sum of the incoming flows is decreased by $\alpha_T(\lambda_v - 1)$. Therefore, the net outgoing flow of every $v \in \mathcal{V} \setminus \{r\}$ is decreased by α_T. Furthermore, since the net outgoing flow of every non-designated node is the same (1) at the beginning, and they are decreased by the same amount (α_T) in every iteration, they are decreased to zero at once.

Now, we can prove the correctness of the algorithm. First, we have to show that the resulting $\boldsymbol{\alpha}$ is indeed a distribution, i.e., $\sum_{T \in \mathcal{T}} \alpha_T = 1$ and $\alpha_T \geq 0$ for every $T \in \mathcal{T}$. This is evident, as the amount of flow consumed by r is decreased by $\alpha_T(|\mathcal{V}| - 1)$ in every iteration when $\sum_{T \in \mathcal{T}} \alpha_T$ in increased by α_T, and the amount of flow consumed is $|\mathcal{V}| - 1$ at the beginning and zero after the algorithm has finished.

Second, we have to show that, for every edge e, $\sum_T \alpha_T \lambda(T, e) \leq f(e)$. At the beginning, the amount of flow along a given edge e is $f(e)$. In every iteration, the flow along the edge is decreased by $\alpha_T \lambda(T, e)$, but it is never decreased to an amount less than zero. Consequently, $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq f(e)$ has to hold.

Finally, we have to show that the algorithm terminates after a finite number of iterations. In every iteration, the flow along at least on edge (that is, along every edge for which $\frac{f(e)}{\lambda(T,e)}$ attains the minimum) is decreased from a positive amount to zero. Since there are a finite number of edges, the algorithm terminates after a finite number of iterations. □

Based on Lemma 2, I provide a polynomial-size characterization for the polyhedron $P_{\mathbf{\Lambda}}$ and its blocker $bl(P_{\mathbf{\Lambda}})$.

Theorem 3. *Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with designated node r, and let $\mathbf{\Lambda}$ be the usage (or loss) matrix of the All-to-One communication model. Then, the polyhedron $P_{\mathbf{\Lambda}}$ can be characterized as*

$$P_{\mathbf{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists f : \mathcal{E} \mapsto \mathbb{R}_{\geq 0} \Bigg(\quad \forall e \in \mathcal{E} : \; f(e) \leq x_e \right. \\ \left. \wedge \;\; \forall v \in \mathcal{V} \setminus \{r\} : \sum_{(v,u)\in\mathcal{E}} f((v,u)) - \sum_{(u,v)\in\mathcal{E}} f((u,v)) = 1 \Bigg) \right\} , \tag{2.33}$$

and its blocker $bl(P_{\mathbf{\Lambda}})$ can be characterized as

$$bl(P_{\mathbf{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \pi \in \mathbb{R}^{|\mathcal{V}|-1} \left(\sum_{v\in\mathcal{V}} \pi_v \geq 1 \;\wedge\; \forall e = (u,v) : \pi_u - \pi_v \leq y_e \right) \right\} , \tag{2.34}$$

where $\pi_r \equiv 0$ by definition to simplify the equation.

Proof. First, I prove Equation (2.33) in two steps:

- $P_{\mathbf{\Lambda}} \supseteq$ Right-hand side (RHS) of Equation (2.33): Notice that the constraints on f in the RHS side of Equation (2.33) actually describe an uncapacitated multi-source network flow: at each node $v \in \mathcal{V} \setminus \{r\}$ (i.e., each non-designated node), the difference between the sum of the outgoing flows $\sum_{(v,u)\in\mathcal{E}} f((v,u))$ and the sum of the incoming flows $\sum_{(u,v)\in\mathcal{E}} f((u,v))$ is 1. In other words, each non-designated node produces an amount of 1. From the conservation of flows, it follows that the designated node r has to consume an amount of $|\mathcal{V}| - 1$. Therefore, a vector $\boldsymbol{x}$ is an element of the RHS if and only if there exists a feasible uncapacitated multi-source flow f such that $\forall e \in \mathcal{E} : f(e) \leq x_e$ and every $v \in \mathcal{V} \setminus \{r\}$ is a source producing an amount of 1, while r is a sink consuming an amount of $|\mathcal{V}| - 1$. From Lemma 2, it follows readily that every element of the RHS has to be an element of the polyhedron $P_{\mathbf{\Lambda}}$, which proves $P_{\mathbf{\Lambda}} \supseteq$ RHS of Equation (2.33).

- $P_{\mathbf{\Lambda}} \subseteq$ RHS of Equation (2.33): We have to show that, for every element $\boldsymbol{x}$ of the polyhedron $P_{\mathbf{\Lambda}}$, there exists a feasible multi-source flow f such that $\forall e \in \mathcal{E} : \; f(e) \leq x_e$ and every $v \in \mathcal{V} \setminus \{r\}$ is a source producing an amount of 1, while r is a sink consuming an amount of $|\mathcal{V}| - 1$. Let $\boldsymbol{x}$ be an arbitrary element of the polyhedron $P_{\mathbf{\Lambda}}$, and let $\boldsymbol{\alpha}$ be a distribution over the set of arborescences such that $\boldsymbol{\alpha}\mathbf{\Lambda} \leq \boldsymbol{x}$. I claim that $f(e) := \sum_{T\in\mathcal{T}} \alpha_T \lambda(T, e)$ is a feasible multi-source flow satisfying the above constraints.

  For any non-designated node $v \in \mathcal{V} \setminus \{r\}$ and arborescence T, the arborescence contains exactly one outgoing edge of v and zero or more incoming edges. If we remove the outgoing edge of v, then all those nodes will be disconnected from r which would be disconnected if we removed the incoming edges of v. Since we also disconnect v when we remove its outgoing edge, we have that the sum of $\lambda(T, e)$ over every incoming edge e is equal to $\lambda(T, e_{outgoing}) - 1$, where $e_{outgoing}$ is the outgoing edge. Furthermore, we have by definition that $\lambda(T, e) = 0$ for every $e \notin T$. Hence, we have

$$\sum_{(v,u)\in\mathcal{E}} \lambda(T,(v,u)) - \sum_{(u,v)\in\mathcal{E}} \lambda(T,(u,v)) \tag{2.35}$$
$$= \lambda(T, e_{outgoing}) - (\lambda(T, e_{outgoing}) - 1) \tag{2.36}$$
$$= 1 \; . \tag{2.37}$$

  Since this holds for every arborescence, we have that the net outgoing flow (i.e., the production)

of a non-designated node v is

$$\sum_{(v,u)\in\mathcal{E}} f((v,u)) - \sum_{(u,v)\in\mathcal{E}} f((u,v)) \tag{2.38}$$

$$= \sum_{(v,u)\in\mathcal{E}} \sum_{T\in\mathcal{T}} \alpha_T \lambda(T,(v,u)) - \sum_{(u,v)\in\mathcal{E}} \sum_{T\in\mathcal{T}} \alpha_T \lambda(T,(u,v)) \tag{2.39}$$

$$= \sum_{T\in\mathcal{T}} \alpha_T \left(\sum_{(v,u)\in\mathcal{E}} \lambda(T,(v,u)) - \sum_{(u,v)\in\mathcal{E}} \lambda(T,(u,v)) \right) \tag{2.40}$$

$$= \sum_{T\in\mathcal{T}} \alpha_T \cdot 1 \tag{2.41}$$

$$= 1 \ . \tag{2.42}$$

Finally, it is easy to see using a similar argument that the consumption of node r is $|\mathcal{V}|-1$, as removing all of its incoming edges disconnects all $|\mathcal{V}|-1$ non-designated nodes. Therefore, f is a feasible multi-source flow satisfying the above constraints.

To prove Equation (2.34), I use Lemma 1. More specifically, I first show how the characterization of Equation 2.33 can be mapped to Equation 2.9, and then use Lemma 1 to find a characterization of the blocker. First, the flow f can be represented by a vector $\boldsymbol{f} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$. Then, $\boldsymbol{S}$ is the identity matrix of size $|\mathcal{E}|$, as we compare $\boldsymbol{x}$ simply to $\boldsymbol{f}$. Second, $\boldsymbol{C}$ is a matrix of size $2(|\mathcal{V}|-1) \times |\mathcal{E}|$, as $|\mathcal{V}|-1$ equalities can be represented by $2(|\mathcal{V}|-1)$ inequalities. To construct matrix $\boldsymbol{C}$, assign integers from 1 to $|\mathcal{V}|-1$ to the non-designated nodes, and assign integers from 1 to $|\mathcal{E}|$ to the edges. Then, for $i \in \{1,\ldots,|\mathcal{V}|-1\}$, the ith row is a vector whose jth element is 1 if the source of edge j is node i, -1 if the target of edge j is node i, and 0 otherwise; and the $(|\mathcal{V}|-1+i)$th row is a vector whose jth element is -1 if the source of edge j is node i, 1 if the target of edge j is node i, and 0 otherwise. Finally, $\boldsymbol{c}$ is a vector consisting of $|\mathcal{V}|$ ones and $|\mathcal{V}|$ negative ones (i.e., $\boldsymbol{c} = [1,\ldots,1,-1,\ldots,-1]'$).

Now, I show using Lemma 1 that the blocker is indeed characterized by Equation 2.34. Since $\boldsymbol{S}$ is the identity matrix, we can omit variable $\boldsymbol{g}$, and characterize the blocker as $\exists \boldsymbol{h}\left(\boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{y} \ \wedge \ \boldsymbol{c}'\boldsymbol{h} \geq 1\right)$, where $\boldsymbol{h} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$. For notational simplicity, let the first half of $\boldsymbol{h}$ be denoted by $\boldsymbol{\pi}^+$, let its second half be denoted by $\boldsymbol{\pi}^-$, and let $\pi^+_r \equiv \pi^-_r \equiv 0$. Then, $\boldsymbol{c}'\boldsymbol{h} \geq 1$ can be written as $\sum_{v\in\mathcal{V}} \pi^+_v - \pi^-_v \geq 1$, and $\boldsymbol{C}'\boldsymbol{h} \leq \boldsymbol{y}$ as $\forall (u,v) \in \mathcal{E} : \pi^+_u - \pi^-_u - \pi^+_v + \pi^-_v \leq y_{(u,v)}$. To further simplify the characterization, I introduce variable $\boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{E}|}$ (notice that this variable is not constrained to be non-negative), let $\pi_v = \pi^+_v - \pi^-_v$ for every non-designated node v, and introduce the notation $\pi_r \equiv 0$. Then, a vector $\boldsymbol{y} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$ is an element of the blocker *iff* $\exists \boldsymbol{\pi} \in \mathbb{R}^{|\mathcal{E}|}\left(\sum_{v\in\mathcal{V}} \pi_v \geq 1 \ \wedge \ \forall (u,v) \in \mathcal{E} : \pi_u - \pi_v \leq y_{(u,v)}\right)$. □

Based on Section 2.3.2, the characterization of $bl(P_{\boldsymbol{\Lambda}})$ presented in the above theorem can readily be used to find the adversary's equilibrium payoff. Furthermore, we can also find an equilibrium strategy profile, as the algorithm presented in the proof of Lemma 2 can be used to compute a mixed operator strategy $\boldsymbol{\alpha}$ from an element $\boldsymbol{x}$ of the polyhedron $P_{\boldsymbol{\Lambda}}$. First, find a feasible *capacitated* multi-source flow f, where the node productions and consumptions are as in Lemma 2 and the capacity of each edge e is x_e. Second, find a mixed operator strategy $\boldsymbol{\alpha}$ (i.e., a distribution over the set of reverse arborescences) using the above algorithm. Then, we have $\sum_{T\in\mathcal{T}} \alpha_T \lambda(T,e) \leq f(e) \leq x_e$ for every e; hence, $\boldsymbol{\alpha}$ is a mixed operator strategy for $\boldsymbol{x}$.[13] Finally, we can show that the running time of this computation is polynomial in the size of the network. Firstly, finding a capacitated flow can be done efficiently using standard algorithms. Secondly, the flow decomposition algorithm has at most $|\mathcal{E}|$ iterations, and each step can be performed in polynomial time. Note that – in order to achieve polynomial running time – we can skip the last step and output only the non-zero coefficients.

2.4.2 Extensions

In the previous section, I have solved a basic model, which assumed a single designated node and limited the adversary to targeting only links. Here, I extend this model and show that the previously presented results hold in the extended model as well.

[13] Recall from Section 2.3.2 that it suffices to find a strategy $\boldsymbol{\alpha}$ for which $\boldsymbol{\alpha}\boldsymbol{\Lambda} \leq \boldsymbol{x}$ holds.

Non-Uniform Node Weights

By defining the loss to be *number* of disconnected nodes, I assumed that each node is equally important or valuable. In practice, however, different nodes can have different value to the operator. For instance, the value of the information collected by the nodes of a sensor network can be non-uniform. To model such non-uniform values, assume that each node v has a given weight $d_v \in \mathbb{R}_{\geq 0}$. Using these weights, the value of the usage (or loss) function $\lambda(T, e)$ can be defined as the sum weight of those nodes that are disconnected from the designated node r in $G[T \setminus \{e\}]$.

To be consistent with the above definition of $\lambda(T, e)$, Lemma 2 has to be reformulated as follows.

Lemma 3. *Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with designated node $r \in \mathcal{V}$, let $\boldsymbol{d}$ be the weights of the nodes, and let $\boldsymbol{\Lambda}$ be the usage (or loss) matrix of the All-to-One communication model. Then, for any $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$, $\boldsymbol{x} \in P_{\boldsymbol{\Lambda}}$ if there exists a feasible uncapacitated multi-source flow f in G such that $\forall e \in \mathcal{E} : f(e) \leq x_e$ and every $v \in \mathcal{V} \setminus \{r\}$ is a source producing an amount of d_v, while r is a sink consuming an amount of $\sum_{v \in \mathcal{V} \setminus \{r\}} d_v$.*

I omit the complete proof since it is almost identical to that of the basic model, and only point out how the proof of the extended model differs. In the proof for the extended model,

- the sum of $\lambda(T, e_{in})$ over all incoming edges e_{in} of a non-designated node v is $\lambda_v - d_v$,
- the net outgoing flow of a non-designated node v is decreased by $\alpha_T d_v$,
- the net outgoing flow of a non-designated node v is d_v at the beginning,
- the amount of flow consumed by the designated node r is decreased by $\alpha_T \sum_{v \in \mathcal{V} \setminus \{r\}} d_v$ in every iteration,
- and the amount of flow consumed by the designated node r is $\sum_{v \in \mathcal{V} \setminus \{r\}} d_v$ in the beginning.

Next, Theorem 3 has to be reformulated as follows.

Theorem 4. *Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with designated node r, let $\boldsymbol{d}$ be the weights of the nodes, and let $\boldsymbol{\Lambda}$ be the usage (or loss) matrix of the All-to-One communication model. Then, the polyhedron $P_{\boldsymbol{\Lambda}}$ can be characterized as*

$$P_{\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists f : \mathcal{E} \mapsto \mathbb{R}_{\geq 0} \Bigg(\quad \forall e \in \mathcal{E} : \; f(e) \leq x_e \right. \\ \left. \wedge \; \forall v \in \mathcal{V} \setminus \{r\} : \sum_{(v,u) \in \mathcal{E}} f((v,u)) - \sum_{(u,v) \in \mathcal{E}} f((u,v)) = d_v \Bigg) \right\}, \tag{2.43}$$

and its blocker $bl(P_{\boldsymbol{\Lambda}})$ can be characterized as

$$bl(P_{\boldsymbol{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \pi \in \mathbb{R}^{|\mathcal{V}|-1} \left(\sum_{v \in \mathcal{V}} d_v \pi_v \geq 1 \;\wedge\; \forall e = (u,v) : \pi_u - \pi_v \leq y_e \right) \right\}, \tag{2.44}$$

where $\pi_r \equiv 0$ by definition to simplify the equation.

Again, instead of repeating the complete proof, I will only point out the main differences. In the proof for the extended model,

- at each node $v \in \mathcal{V} \setminus \{r\}$, the difference between the sum of the outgoing flows $\sum_{(v,u) \in \mathcal{E}} f((v,u))$ and the sum of the incoming flows $\sum_{(u,v) \in \mathcal{E}} f((u,v))$ is d_v; in other words, each non-designated node v produces an amount of d_v,
- the designated node consumes an amount of $\sum_{v \in \mathcal{V} \setminus \{r\}} d_v$,
- the sum of $\lambda(T, e)$ over every incoming edge e is equal to $\lambda(T, e_{outgoing}) - d_v$, where $e_{outgoing}$ is the outgoing edge,
- vector $\boldsymbol{c}$ is the concatenation of $\boldsymbol{d}$ and $-\boldsymbol{d}$ (i.e., $\boldsymbol{c} = [d_1, \ldots, d_{|\mathcal{V}|-1}, -d_1, \ldots, -d_{|\mathcal{V}|-1}]'$),
- and $\boldsymbol{c}'\boldsymbol{h} \geq 1$ can be written as $\sum_{v \in \mathcal{V}} d_v(\pi^+_v - \pi^-_v) \geq 1$.

Multiple Designated Nodes

In practice, an all-to-one network can have more than one designated node. For example, there can be multiple data collection nodes in a sensor network or multiple gateways in an access network. To model multiple designated nodes, assume that a set of designated nodes R is given instead of only a single one, and each non-designated node needs to be connected to only one of these designated nodes. Then, the value of the usage (or loss) function $\lambda(T, e)$ is the sum weight of those nodes from which there is no directed path in $G[T \setminus \{e\}]$ to any of the designated nodes R.

I now show that the problem of solving a game with multiple designated nodes can be reduced to the problem of solving a game with a single designated node. Add a "super"-designated node r^* to the graph, and connect each designated node $r \in R$ to r^* with an edge having infinite attack cost $\mu_{(r,r^*)} \to \infty$.[14] Now, solve the game as before with r^* as the single designated node. Since the attack costs prevent the adversary from targeting the edges between the designated nodes R and the "super"-designated node r, there has to be a path from a node v to the "super"-designated node r^* if there is a path from node v to any of the designated nodes R. Hence, the adversary's equilibrium payoff and her set of equilibrium strategies have to be the same as in the extended model. Based on the above argument, I will restrict my analysis for the remainder of this chapter to modeling a single designated node, and note that results for multiple designated nodes can be obtained easily using the above construction.

Attacks against Nodes

In many scenarios, the adversary is not limited to attacking only links, but can also target the nodes of the network. For example, in a sensor network deployed over an area that is not fenced off (or guarded somehow), it is hard to protect the nodes from physical attacks. To extend the model, assume that the adversary can target both edges and nodes, that is, her pure-strategy set is $\mathcal{E} \cup \mathcal{V}$, and each non-designated node v has an attack cost μ_v. Then, the usage (or loss) function $\lambda : \mathcal{E} \cup \mathcal{V} \to \mathbb{R}_{\geq 0}$ can be defined as follows: for a given element (i.e., edge or node) $e \in \mathcal{E} \cup \mathcal{V}$ and arborescence T, the value of $\lambda(T, e)$ is the sum weight of those nodes from which the path to the designated node r in T contains e.

I now show that the problem of solving a game with attacks against both nodes and edges can be reduced to the problem of solving a game with attacks against only edges. Replace each non-designated node v with two nodes, denoted by v_1 and v_2. For each non-designated node v, let $d_{v_1} = d_v$, let $d_{v_2} = 0$, and add an edge (v_1, v_2) having an attack cost of $\mu_{(v_1,v_2)} = \mu_v$. Finally, replace each edge (u, v) with an edge (u_2, v_1) having the same attack cost as the original edge. Intuitively, an attack against node v in the extended model corresponds to the attack against edge (v_1, v_2) in the constructed network, as they have the same cost, μ_v, and the same loss (i.e., sum weight of disconnected nodes). Similarly, an attack against edge (u, v) in the extended model corresponds to the attack against edge (u_2, v_1) in the constructed network, as they have the same cost, $\mu_{(u,v)}$, and same loss. Then, it is fairly easy to see that the vulnerability of the constructed network is equal to the that of the original network in the extended model. Furthermore, the players' equilibrium strategies are also identical if the above correspondence rules are applied to them. Based on the above argument, I will restrict my analysis for the remainder of this chapter to modeling only attacks against links, and note that results for attacks against both links and nodes can be obtained easily using the above construction.

2.4.3 Graph-Theoretic Metric

In Section 2.4.1, I focused on the computational complexity of solving a network blocking game in the All-to-One communication model and showed how the game can be solved in polynomial-time. In this section, I present a more intuitive, closed-form expression of the adversary's payoff – that is, our vulnerability metric – and show that a previously proposed graph-theoretic robustness metric, called directed graph strength [Cunningham, 1985], is closely related to our metric. Note that this section builds on the results of the preceding one and allows the nodes to have non-uniform weights $\boldsymbol{d}$.

I begin with showing that the extreme points of the blocker correspond to the minimal cuts of the network graph. The importance of the extreme points was discussed in Section 2.2.2 (Theorem 1 in particular). Firstly, the function θ attains its maximum at an extreme point (or at some extreme points) of the blocker; hence, the vulnerability θ_{max} of the graph can be expressed using the set of extreme

[14] In practice, the attack cost $\mu_{(r,r^*)}$ can be a sufficiently large but finite number.

points. Secondly, equilibrium adversarial strategies can be constructed from combinations of extreme points at which the maximum is attained; hence, these critical extreme points can be used to predict the most likely attacks. The following theorem shows that the extreme points of the blocker $bl(P_{\mathbf{\Lambda}})$ in the All-to-One model correspond to minimal cuts in the network graph. Note that, since the graph is directed, so are the minimal cuts, which are "facing" the designated node r.

Theorem 5. *Let $G = (\mathcal{V}, \mathcal{E})$ be a directed graph with a designated node r, let $\boldsymbol{d}$ be the weights of the nodes, and let $\mathbf{\Lambda}$ be the usage (or loss) matrix of the All-to-One communication model. Then, the set of extreme points of the blocker $bl(P_{\mathbf{\Lambda}})$ is $\left\{\boldsymbol{\omega}^C \;\middle|\; C \subseteq \mathcal{E} : C \text{ is a minimal cut of } G\right\}$, where*

$$\omega_e^C = \frac{1_{e \in C}}{\lambda_r(C)} \ , \tag{2.45}$$

where $\lambda_r(C)$ denotes the sum weight of those nodes that are disconnected from r if C is removed from the graph.

Based on the above theorem, the vulnerability of the graph G can be expressed as

$$\begin{aligned}
\theta_{max}(G) &= \max_{\boldsymbol{y} \in bl(P_{\mathbf{\Lambda}})} \theta(\boldsymbol{y}) && (2.46)\\
&= \max_{C \subseteq \mathcal{E} \,:\, C \text{ is a minimal cut of } G} \theta(\boldsymbol{\omega}^C) && (2.47)\\
&= \max_{C \subseteq \mathcal{E} \,:\, C \text{ is a minimal cut of } G} \frac{1}{\sum_{e \in \mathcal{E}} \frac{1_{e \in C}}{\lambda_r(C)}} \left(1 - \sum_{e \in \mathcal{E}} \mu_e \frac{1_{e \in C}}{\lambda_r(C)}\right) && (2.48)\\
&= \max_{C \subseteq \mathcal{E} \,:\, C \text{ is a minimal cut of } G} \frac{\lambda_r(C)}{|C|} \left(1 - \sum_{e \in C} \frac{\mu_e}{\lambda_r(C)}\right) && (2.49)\\
&= \max_{C \subseteq \mathcal{E} \,:\, C \text{ is a minimal cut of } G} \frac{\lambda_r(C)}{|C|} - \sum_{e \in C} \frac{\mu_e}{|C|} \ . && (2.50)
\end{aligned}$$

Intuitively, this closed-form expression says that an attacker should focus her attack on (combinations of) minimal cuts that maximize the ratio between the weight of disconnected nodes and the number of edges removed and - at the same time - minimize her average attack cost. Notice that, since $\boldsymbol{\omega}^C$ is uniform over its support, the first term (i.e., $\frac{\lambda_r(C)}{|C|}$) is actually the expected weight of nodes disconnected after the attack corresponding to $\boldsymbol{\omega}^C$. Theoretically, this formulation can also be used to compute vulnerability by iterating over the set of minimal cuts and evaluating the value of θ for each minimal cut. However, due to the very large number of minimal cuts, this is approach is rather impractical.

Proof. First, I show that every element of the blocker $bl(P_{\mathbf{\Lambda}})$ can be expressed as a sum of a non-negative vector and a convex linear combination of the elements of $\left\{\boldsymbol{\omega}^C \;\middle|\; C \subseteq \mathcal{E} : C \text{ is a minimal cut of } G\right\}$. Let $\boldsymbol{y}$ be an arbitrary element of the blocker $bl(P_{\mathbf{\Lambda}})$. Then, let $\pi : \mathcal{V} \to \mathbb{R}_{\geq 0}$ be the potential function defined such that, for every $v \in \mathcal{V}$, the potential $\pi(v)$ is the cost of the minimum cost path from v to r, where the cost of each edge e is y_e. First, I show that $\sum_{v \in \mathcal{V}} d_v \pi(v) \geq 1$. Let T be an arbitrary reverse arborescence in which every path from a non-designated node to r is a minimum cost path.[15] Then,

[15] One can construct such a reverse arborescence T easily by, for each non-designated node v, adding the first edge of a minimum cost (v, r) path to T.

we have

$$\sum_{v\in\mathcal{V}} d_v \pi(v) \tag{2.51}$$

$$= \sum_{v\in\mathcal{V}} d_v \left(\sum_{e\in \text{ edges of the } (v,r) \text{ path in } T} y_e \right) \tag{2.52}$$

$$= \sum_{e\in\mathcal{E}} \underbrace{\sum_{v\in\mathcal{V} \text{ such that } e\in \text{ edges of the } (v,r) \text{ path in } T} d_v\, y_e}_{=\lambda(T,e)} \tag{2.53}$$

$$= \sum_{e\in\mathcal{E}} \lambda(T,e) y_e \tag{2.54}$$

$$\geq 1\ . \tag{2.55}$$

First, Equality (2.52) is true as every path in T is a minimum cost path. In Equality (2.53), I simply changed the order of summation. Equality (2.54) follows from the definition of λ in the All-to-One model. Finally, Equality (2.55) holds because $\boldsymbol{y}$ was chosen to be an element of the blocker.

Based on this property of the potential function π, I propose the following algorithm for expressing $\boldsymbol{y}$ as a convex linear combination of minimal cut vectors.

1. Let u be a node with maximum potential (i.e., a node at which the potential function π attains its maximum).
2. Let U be the set of nodes that can be reached from u using zero-cost paths (i.e., the set of nodes to which there exists a directed path from u consisting of only edges not in the support of $\boldsymbol{y}$).
3. Let C be the set of edges leaving U (i.e., the set of edges whose sources are in U and whose targets are not in U).
4. Let the coefficient ξ_C of $\boldsymbol{\omega}^C$ be $\min_{e\in C} \frac{y_e}{\lambda_r(C)}$.
5. Let $\boldsymbol{y} := \boldsymbol{y} - \xi_C \boldsymbol{\omega}^C$ and recalculate the values of π accordingly.
6. If there is a node with non-zero potential, then continue from Step 1.

First, observe that the set of edges C is a minimal cut by definition. Next, I show that the potential of every node cut off by the removal of C[16] has to be equal to $\pi(u)$. Firstly, no node can have a higher potential since $\pi(u)$ is the maximum by definition. Secondly, no node v in U can have a lower potential; otherwise, there would exist a (v,r) path whose cost is less than $\pi(u)$. But this would lead to a contradiction, since the concatenation of a zero-cost (u,v) path and a minimum cost (v,r) path would have a cost that is lower than $\pi(u)$, which would contradict that the potential of u is $\pi(u)$. Thirdly, for the remaining nodes (i.e., those that are cut off by the removal of C but are not an element of U), every path to r has to include at least one node in U; otherwise, they would not be cut off by the removal of C. Then, it follows readily that they cannot have a potential lower than $\pi(u)$, as the cost of any path to r has to be at least $\pi(u)$. Now, notice that this uniformity of the potential function also implies that there has to exist a zero-cost path to an element of U from every node cut off by the removal of C. Hence, if we decrease the cost y_e of every edge e in C uniformly, then we also decrease the potential of every node cut off by the removal of C uniformly.

Next, I show that the algorithm terminates after a finite number of iterations. In Step 5, the cost y_e of an edge e is decreased to zero (i.e., the edge – or edges – where $\xi_C = \frac{y_e}{\lambda_r(C)}$ in Step 4). Hence, in every iteration, the number of edges with positive costs is decreased by one. Consequently, after $|\mathcal{E}|$ iterations, the cost of every edge would be zero, which obviously implies that the potential of every node is also zero. Therefore, the algorithm terminates after at most $|\mathcal{E}|$ iterations.

Now, I prove the correctness of the algorithm. We readily have that $\sum_{C \text{ in minimal cuts}} \boldsymbol{\omega}^C = \boldsymbol{y}$ from the definition of the algorithm (see Step 5). It remains to show that $\boldsymbol{\xi}$ are indeed the coefficients of a

[16] Note that this set can be a strict superset of U.

convex combination, i.e., that $\sum_{C \text{ in minimal cuts}} \xi_C = 1$. In every iteration, the potential of the nodes cut off by the removal of C is decreased by $\xi_C \frac{1}{\lambda_r(C)}$, as the costs of the edges between this set and the remainder of the graph are decreased by $\xi_C \frac{1}{\lambda_r(C)}$. Hence, the sum $\sum_{v \in \mathcal{V}} d_v \pi(v)$ is decreased by $\lambda_r(C) \xi_C \frac{1}{\lambda_r(C)} = \xi_C$. Furthermore, this sum is never decreased to a negative value by definition, and it is equal to 1 at the beginning. Consequently, $\sum_{C \text{ in minimal cuts}} \xi_C = 1$ has to hold. Therefore, every element of the blocker $bl(P_{\mathbf{\Lambda}})$ can be expressed as the sum of a non-negative vector and a convex linear combination of the elements of $\left\{ \boldsymbol{\omega}^C \;\middle|\; C \subseteq \mathcal{E} : C \text{ is a minimal cut of } G \right\}$.

It remains to show that the set $\left\{ \boldsymbol{\omega}^C \;\middle|\; C \subseteq \mathcal{E} : C \text{ is a minimal cut of } G \right\}$ is conically independent (i.e., no element of the set can be expressed as a conic combination of the remaining elements) and that every element of the set if an element of the blocker. The first one follows readily from the definition of minimal cuts. To prove the second one, it suffices to show that each $\boldsymbol{\omega}^C$ blocks every row $\boldsymbol{\lambda}_T$ of the matrix $\mathbf{\Lambda}$. Let C be an arbitrary minimal cut, and let T be an arbitrary reverse arborescence rooted at r. Now, there has to be at least one edge c in C whose removal from T disconnects $\lambda_r(C)$ nodes. Consequently, we have $\lambda(T, c) \geq \lambda_r(C)$. Then,

$$\sum_{e \in \mathcal{E}} \lambda(T, e) \omega_e^C \tag{2.56}$$

$$\geq \lambda(T, c) \omega_c^C \tag{2.57}$$

$$\geq \lambda_r(C) \omega_c^C \tag{2.58}$$

$$= \lambda_r(C) \frac{1}{\lambda_r(C)} = 1 \; . \tag{2.59}$$

Therefore, $\boldsymbol{\omega}^C$ is an element of the blocker. □

Directed Graph Strength

In this section, I focus on a special case of the All-to-One network blocking game where the attack costs μ_e are all zero. This special case is particularly interesting because – in this case – our game-theoretic robustness metric is equivalent to the special case of a metric previously proposed by Cunningham in [Cunningham, 1985] based on purely graph-theoretical considerations.

Formally, assume that $\boldsymbol{\mu} = \mathbf{0}$ for the remainder of this section. Then, from Equation (2.50), we have that the vulnerability of graph G is

$$\theta_{max}(G) = \max_{C \subseteq \mathcal{E} : C \text{ is a minimal cut of } G} \frac{\lambda_r(C)}{|C|} - \sum_{e \in C} \frac{0}{|C|} \tag{2.60}$$

$$= \max_{C \subseteq \mathcal{E} : C \text{ is a minimal cut of } G} \frac{\lambda_r(C)}{|C|} \; . \tag{2.61}$$

Equivalently, the robustness of graph G is

$$\frac{1}{\theta_{max}(G)} = \min_{C \subseteq \mathcal{E} : C \text{ is a minimal cut of } G} \frac{|C|}{\lambda_r(C)} \; . \tag{2.62}$$

Now, consider the robustness metric proposed in [Cunningham, 1985], called directed graph strength.

Definition 3 (Directed Graph Strength)**.** Let G be a directed graph with a designated node r. Then, the *directed strength* of G, denoted by $\pi(G)$, is

$$\pi(G) = \min_{F \subseteq \mathcal{E}} \frac{\sum_{e \in F} s(e)}{\lambda_r(F)} \; , \tag{2.63}$$

where $s(e)$ measures the cost of attacking edge e.

It is fairly easy to see that the above maximum is attained for some minimal cut [Cunningham, 1985]. Hence, when $s(e) \equiv 1$, we have $\theta_{max}^{-1}(G) = \pi(G)$.

It is interesting two compare how the two metrics incorporate attack costs. In our game-theoretic metric, the adversary is trying to maximize the *difference* between her expected reward and her expected costs (see Equation (2.50)). In directed graph strength, the adversary is trying to maximize the *ratio* between her reward and her costs (see above definition). In other words, she prefers attacks maximizing her expected net reward in the former, and attacks maximizing her benefit to cost ratio in the latter.

2.5 All-to-All Communications Model with Linear Usage

The All-to-One communication model introduced in the previous section is well-suited for modeling a wide range of networks, where the nodes have to remain connected to a (set of) designated nodes, such as sensor or access networks. However, there are many networks where such designated nodes do not exist. For instance, in a local area network whose purpose is to enable the users to collaborate with each other, the nodes have to remain connected to each other. As another example, in a backbone network that is connecting multiple subnetworks, the nodes again have to be connected to each other. To study the robustness of these networks, a different communication model is needed. In this section, I propose a communication model for such networks.

In [Gueye et al., 2010], the authors introduced a communication model, which I call here the All-to-All model with constant loss, for modeling networks where the nodes have to be connected to each other. In this model, the operator uses a collection of links T forming a spanning tree (i.e., a minimal connected or – equivalently – a maximal loop-free set of edges). Hence, the set of feasible collections $\mathcal{T}$ is the set of all spanning trees. This choice for the operator's strategy space can be motivated by, for example, local area networks, where a spanning tree is created between the bridges by disabling some of the links, leaving a single path between any pair of nodes.

Finding a realistic usage function $\lambda(T, e)$, on the other hand, is a more challenging task. However, instead of trying to estimate the usage of a link, one can also define $\lambda(T, e)$ by estimating the amount of loss caused by the removal of the link. In other words, instead of focusing on the usage of a link when it is not attacked, one can focus on the loss sustained when the link is attacked. In [Gueye et al., 2010], a simple, constant loss function was proposed for this purpose. This loss function can be expressed as $\lambda(T, e) = 1_{e \in T}$, and the resulting communication model can be solved in polynomial-time. However, the assumption that the loss is constant, regardless of how disconnected the networks becomes as as a result of the attack, is obviously a very rough estimation. In [Gueye et al., 2012], a number of new loss functions were introduced, which were based on previously proposed models for estimating the value of a network. The idea behind these loss functions is that the loss can be estimated as the decrease in network value. Unfortunately, there is no known polynomial-time algorithm for solving these communication models (except for the constant loss model).

In this section, I propose a new communication model based on a linear loss (or usage) function, which can be solved efficiently. This linear loss function is defined as follows: for a given spanning tree T and edge e, the loss $\lambda(T, e)$ is the *number of nodes in the smaller component of* $G[T \setminus \{e\}]$. The rationale behind this function is that, the more nodes become separated due to the attack, the higher the loss is. For example, if only a single node is separated from the rest of the network, then the large majority of nodes can still communicate with each other and the loss can be considered minor. On the other hand, if the network is cut in half by the attack, then the majority of the node pairs become disconnected and the loss can be considered serious. In the following subsections, I compare the proposed linear loss function to the previously introduced All-to-All loss functions. Then, in Section 2.5.2, I provide a polynomial-size linear characterization for the the All-to-All communication model with linear loss. Finally, in Section 2.5.3, I show that the resulting vulnerability metric is closely related to the Cheeger constant, a well-known graph-theoretic metric.

2.5.1 Comparison with Other All-to-All Models

Figure 2.1 compares the proposed linear loss function to the functions introduced in [Gueye et al., 2012] and the constant loss function (denoted by GWA in the figure). The comparison is based on a network consisting of 60 nodes. The horizontal axis shows the sizes of the components of the graph resulting from the attack. Extreme values 0 and 60 correspond to intact networks (i.e., when $e \notin T$). Values $0 < n < 60$ correspond to a damaged network consisting of two components of sizes n and $60 - n$. The vertical axis shows the payoff of the adversary or – equivalently – the loss of the operator. The previously proposed loss functions are "normalized" in the manner described in [Gueye et al., 2012]: each loss function is divided by the value of the intact network, which is determined by the corresponding network value function. The proposed linear loss function is not based on a network value function, but since each node being separated causes a loss of value 1, we can establish that the value of the intact network is the number of nodes $|\mathcal{V}|$. Therefore, the linear loss function is "normalized" by dividing it with the constant coefficient $|\mathcal{V}|$. The normalization allows us to make an unbiased comparison between different

loss functions.

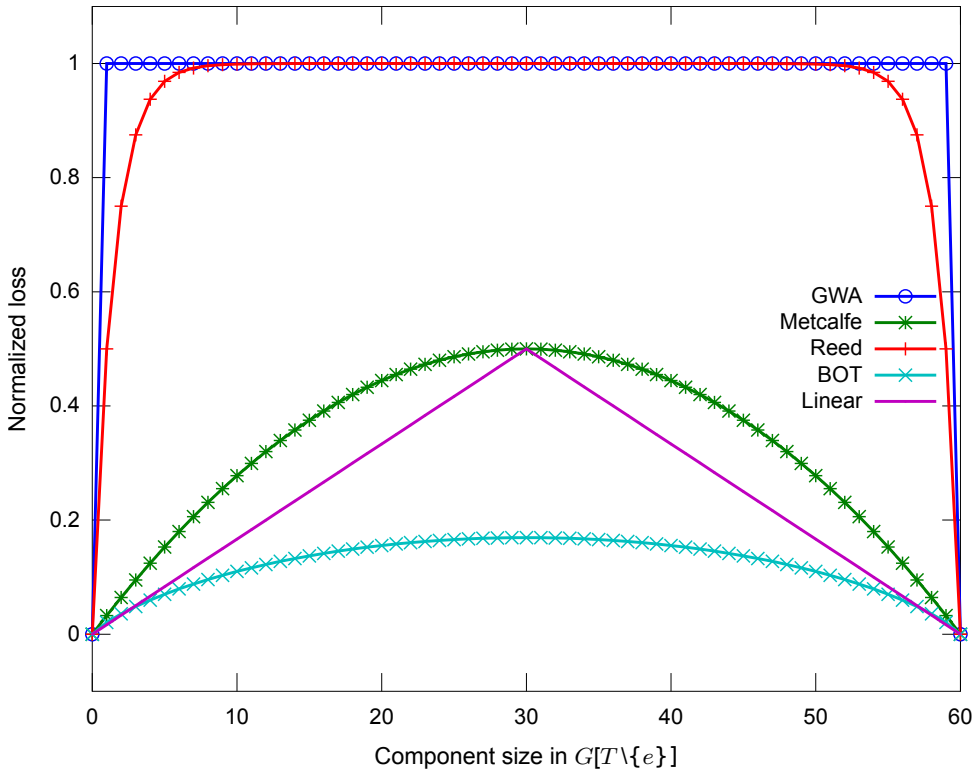


Figure 2.1: Comparison of various loss functions.

The figure shows that the linear loss function is bounded by the previously proposed loss functions, Metcalfe and BOT. The Metcalfe function measures a special quadratic loss and is an upper bound of the linear function. The BOT function measures a special logarithmic loss and is a lower bound of the linear function. For the exact definitions and a discussion of these functions, I refer the reader to [Gueye et al., 2012]. We can conclude that the linear loss function is at least as realistic as the previously proposed functions. Please recall that, to the best of my knowledge, there are no polynomial-time algorithms known to compute the equilibria or the optimal strategies of the games based on these functions, except for the constant loss function.

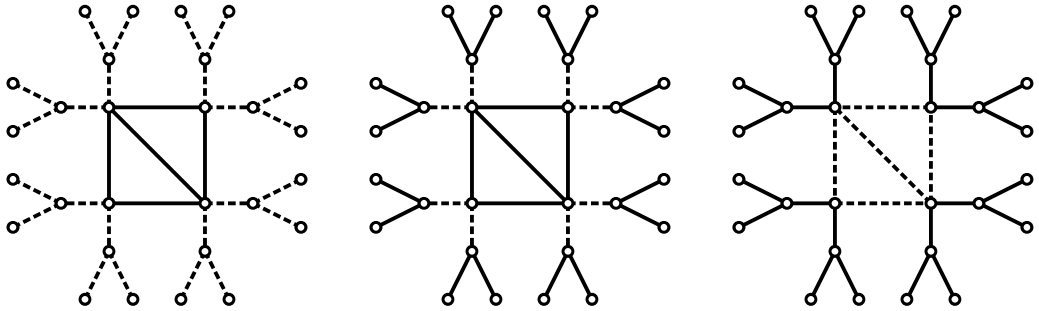
Sets of Critical Edges

Besides quantifying the robustness of network topologies, network blocking games can also be used to identify critical edges that are likely to be attacked. Formally, an edge is called *critical* if it is in the support of an optimal adversarial strategy [Gueye et al., 2010]. In this subsection, I make a comparison between the sets of critical edges identified using various loss functions.

In [Gueye et al., 2012], the proposed loss functions were compared based on the example network shown in Figure 2.2 with $\boldsymbol{\mu} = \mathbf{0}$. For the sake of consistency, I use the same network for comparison. The sets of critical edges identified using the previously proposed loss functions are taken from [Gueye et al., 2012].

Figure 2.2a shows the set of critical edges identified using the constant loss function $1_{e \in T}$ (denoted by GWA). The critical set consists exclusively of bridges, edges whose removal disconnects the network. This can be explained by the fact that the constant loss function does not take into account the magnitude of the damage. Therefore, the adversary maximizes solely the probability of hitting the spanning tree, regardless of the expected number of nodes cut off.

Figure 2.2b shows the set of critical edges identified using the Reed loss function. The set is similar to the one identified using the constant loss function, but contains only those bridges that cut off more than one node. This result is consistent with Figure 2.1, which also shows that the Reed and the constant



(a) GWA (b) Reed (c) BOT, Metcalfe and Linear

Figure 2.2: Sets of critical edges for various loss functions with $\boldsymbol{\mu} = \mathbf{0}$. Critical edges are represented by dashed lines.

loss functions are similar, but the Reed function also takes the magnitude of the damage into account to some extent.

Finally, Figure 2.2c shows the set of critical edges identified using the BOT, Metcalfe, and the linear loss functions. Again, the fact that these three functions result in the same set is consistent with the earlier comparison based on Figure 2.1, which also showed that these functions are similar.

2.5.2 Solving the Game

In this subsection, I provide a polynomial-size linear characterization of the polyhedron $P_{\boldsymbol{\Lambda}}$ associated with the All-to-All communication model with linear loss. Combined with the results of Section 2.3.2, this characterization allows us to compute the adversary's equilibrium payoff and an equilibrium adversarial strategy in polynomial time. Furthermore, the proof of the following lemma (Lemma 4) can be used to compute a mixed operator strategy from an element of the polyhedron $P_{\boldsymbol{\Lambda}}$ in polynomial time; hence, we can find an equilibrium strategy profile efficiently.

Lemma 4. *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph, and let $\boldsymbol{\Lambda}$ be the usage (or loss) matrix of the All-to-All model with linear usage. Then, for any $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$, $\boldsymbol{x} \in P_{\boldsymbol{\Lambda}}$ if there exists a distribution $\boldsymbol{a}$ over the set of nodes (i.e., $\boldsymbol{a} \in \mathbb{R}^{|\mathcal{V}|}_{\geq 0}$ and $\boldsymbol{a}'\mathbf{1} = 1$) and a set of $|\mathcal{V}|$ feasible uncapacitated multi-source flows, denoted by $\{f^r \mid r \in \mathcal{V}\}$, satisfying the following constraints:*

- *for every edge $e = \{u, v\} \in \mathcal{E}$, $\sum_{r \in \mathcal{V}} f^r(u, v) + f^r(v, u) \leq x_e$ and,*
- *in flow f^r, node r is a sink consuming an amount of $a_r(|\mathcal{V}| - 1)$, while every other node is a source producing an amount of a_r.*

Proof. We have to show that the lemma holds for every $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$. Let $\boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}$ be an arbitrary vector for which a distribution $\boldsymbol{a}$ and a set of feasible multi-source flows $\{f^r \mid r \in \mathcal{V}\}$ satisfying the constraints of the lemma exist. We have to show that there is a distribution $\boldsymbol{\alpha}$ over the set of spanning trees such that $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq x_e$ holds for every edge $e \in \mathcal{E}$. I prove this by showing that there exists a distribution $\boldsymbol{\alpha}$ such that $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq \sum_{r \in \mathcal{V}} f^r(u, v) + f^r(v, u)$ for every $e = \{u, v\} \in \mathcal{E}$. That is, I show that any set of feasible multi-source flows $\{f^r \mid r \in \mathcal{V}\}$ satisfying the constraints of the lemma can be expressed as a convex combination of weighted spanning trees. Let $\lambda_r(T, e)$ denote the number of nodes from which there is no path to r in $G[T \setminus e]$.[17] Then, the claim can be proved by the following algorithm.

1. For every node $r \in \mathcal{V}$,
   (a) find a combination $\boldsymbol{\alpha}^r$ of spanning trees that satisfies $\mathbf{1}'\boldsymbol{\alpha}^r = a_r$ and $\sum_{T \in \mathcal{T}} \alpha^r_T \lambda(T, e) \leq f^r(u, v) + f^r(v, u)$ for every $e = \{u, v\} \in \mathcal{E}$ using the algorithm from the proof of Lemma 2.
2. Let $\alpha_T := \sum_{r \in \mathcal{V}} \alpha^r_T$ for every spanning tree.

[17] In other words, let $\lambda_r(T, e)$ denote the value of $\lambda(T, e)$ in the All-to-One communication model with designated node r.

First, I elaborate on how the algorithm from the proof of Lemma 2 can be used in Step 1. In a nutshell, we first have to create a directed graph and a directed flow, then find a combination of arborescences, and finally convert it to a combination of spanning trees.

In more detail, first replace each edge $\{u, v\}$ in G with two directed edges, (u, v) and (v, u), facing opposite directions, and let $f((u, v))$ be $f^r(u, v)$ for each directed edge. Since the sum outgoing and incoming flow of each node remains the same, we have that f is a feasible multi-source flow in G such that every $v \in \mathcal{V} \setminus \{r\}$ is a source producing an amount of a_r, while r is a sink consuming an amount of $a_r(|\mathcal{V}| - 1)$. Consequently, we can run the algorithm to find a combination of arborescences. Now, since there is only one orientation for each set of undirected edges that forms an arborescence rooted at r, there is a one-to-one correspondence between the set of arborescences and the set of spanning trees. Hence, by omitting the orientation, we can use the output of the algorithm readily as $\boldsymbol{\alpha}^r$. Furthermore, we have from the proof of Lemma 2 that the $\boldsymbol{\alpha}^r$ output by the algorithm satisfies $\mathbf{1}'\boldsymbol{\alpha}^r = a_r$, as the consumption of the sink node r is a_r. It remains to show that $\sum_{T \in \mathcal{T}} \alpha_T^r \lambda(T, e) \leq f^r(u, v) + f^r(v, u)$ for every $e = \{u, v\} \in \mathcal{E}$. From the proof of Lemma 2, we have that $\sum_{T \in \text{arborescences}} \alpha_T^r \lambda_r(T, e) \leq f(e)$ for each directed edge e. From the construction of the directed graph we also have that $\sum_{T \in \mathcal{T}} \alpha_T^r \lambda_r(T, \{u, v\}) = \sum_{T \in \text{arborescences}} \alpha_T^r (\lambda_r(T, (u, v)) + \lambda_r(T, (v, u)))$ for each undirected edge $\{u, v\}$, as both orientations of the edge can be used by the arborescences. Therefore, we have

$$\begin{aligned}
&\sum_{T \in \mathcal{T}} \alpha_T^r \lambda_r(T, \{u, v\}) && (2.64)\\
=&\sum_{T \in \text{ arborescences}} \alpha_T^r \left(\lambda_r\left(T, (u, v)\right) + \lambda_r\left(T, (v, u)\right)\right) && (2.65)\\
\leq& f((u, v)) + f((v, u)) && (2.66)\\
=& f^r(u, v) + f^r(v, u)\ . && (2.67)
\end{aligned}$$

Finally, I show that $\boldsymbol{\alpha}$ is a distribution satisfying $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq x_e$ for every $e \in \mathcal{E}$. First, we have from the algorithm that

$$\begin{aligned}
\sum_{T \in \mathcal{T}} \alpha_T &= \sum_{T \in \mathcal{T}} \sum_{r \in \mathcal{V}} \alpha_T^r && (2.68)\\
&= \sum_{r \in \mathcal{V}} \sum_{T \in \mathcal{T}} \alpha_T^r && (2.69)\\
&= \sum_{r \in \mathcal{V}} a_r && (2.70)\\
&= 1\ , && (2.71)
\end{aligned}$$

which proves that $\boldsymbol{\alpha}$ is indeed a distribution. Second, I show that $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq \sum_{r \in \mathcal{V}} f^r(u, v) + f^r(v, u)$ for every $e = \{u, v\} \in \mathcal{E}$. I prove this by showing that $\lambda(T, e) \leq \lambda_r(T, e)$ for every $T \in \mathcal{T}$ and $e \in \mathcal{E}$. Let e and T be an arbitrary edge and spanning tree. Then, $\lambda(T, e)$ is the number of nodes in the smaller component of $G[T \setminus e]$, while $\lambda_r(T, e)$ is the number of nodes that are separated from r in $G[T \setminus e]$. In other words, $\lambda_r(T, e)$ is the number of nodes in the component that does not contain r. Since the number of nodes in the smaller component is obviously less than or equal to the number of nodes in the component that does not contain r, we have $\lambda(T, e) \leq \lambda_r(T, e)$. Then, for every edge $e = \{u, v\}$,

$$\begin{aligned}
\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) =& \sum_{r \in \mathcal{V}} \sum_{T \in \mathcal{T}} \alpha_T^r \lambda(T, e) && (2.72)\\
\leq& \sum_{r \in \mathcal{V}} \sum_{T \in \mathcal{T}} \alpha_T^r \lambda_r(T, e) && (2.73)\\
\leq& \sum_{r \in \mathcal{V}} f^r(u, v) + f^r(v, u) && (2.74)\\
\leq& x_e\ . && (2.75)
\end{aligned}$$

Therefore, $\boldsymbol{x} \in P_{\boldsymbol{\Lambda}}$ holds. □

Based on Lemma 4, I provide a polynomial-size characterization for the polyhedron $P_{\boldsymbol{\Lambda}}$ and its blocker $bl(P_{\boldsymbol{\Lambda}})$.

Theorem 6. *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph, and let $\mathbf{\Lambda}$ be the usage (or loss) matrix of the All-to-All model with linear usage. Then, the polyhedron $P_{\mathbf{\Lambda}}$ can be characterized as*

$$P_{\mathbf{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \,\middle|\, \exists \boldsymbol{a} \in \mathbb{R}^{|\mathcal{V}|}_{\geq 0}, \left\{ f^r : \mathcal{E} \mapsto \mathbb{R}^2_{\geq 0} \,\middle|\, r \in \mathcal{V} \right\} \left(\begin{array}{l} \boldsymbol{a}'\mathbf{1} = 1 \\ \quad \wedge \ \forall e = \{u,v\} \in \mathcal{E} : \sum_{r \in \mathcal{V}} f^r(u,v) + f^r(v,u) \leq x_e \\ \quad \wedge \ \forall r \in \mathcal{V}, v \in \mathcal{V} \setminus \{r\} : \sum_{\{v,u\} \in \mathcal{E}} f^r(v,u) - f^r(u,v) = a_r \end{array} \right) \right\}, \tag{2.76}$$

and its blocker $bl(P_{\mathbf{\Lambda}})$ can be characterized as

$$bl(P_{\mathbf{\Lambda}}) = \left\{ \boldsymbol{y} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\} \left(\begin{array}{l} \forall r \in \mathcal{V} : \sum_{v \in \mathcal{V}} \pi_r(v) \geq 1 \\ \quad \wedge \ \forall r \in \mathcal{V}, e = \{u,v\} \in \mathcal{E} : |\pi_r(v) - \pi_r(u)| \leq y_e \end{array} \right) \right\}, \tag{2.77}$$

where $\pi_r(r) \equiv 0$ for every $r \in \mathcal{V}$ by definition to simplify the equation. Note that the constraint $|\pi_r(v) - \pi_r(u)| \leq y_e$ is equivalent to $\pi_r(v) - \pi_r(u) \leq y_e \wedge \pi_r(u) - \pi_r(v) \leq y_e$; hence, the above characterization can actually be expressed using a linear formulas only, and the absolute value function is used solely as a notational simplification.

Proof. First, I prove Equation (2.76) in two steps:

- $P_{\mathbf{\Lambda}} \supseteq$ Right-hand side (RHS) of Equation (2.76): Notice that the constraints on $\{f^r \,|\, r \in \mathcal{V}\}$ in the RHS side of Equation (2.76) actually describe a set of uncapacitated multi-source network flows: in flow f^r, at each node $v \in \mathcal{V} \setminus \{r\}$, the difference between the sum of the outgoing flows $\sum_{\{v,u\} \in \mathcal{E}} f^r(v,u)$ and the sum of the incoming flows $\sum_{\{u,v\} \in \mathcal{E}} f^r(u,v)$ is a_r. In other words, each node $v \neq r$ produces an amount of a_r. From the conservation of flows, it follows that node r has to consume an amount of $a_r(|\mathcal{V}| - 1)$. Therefore, a vector $\boldsymbol{x}$ is an element of the RHS if and only if there exists a distribution $\boldsymbol{a}$ and a set of uncapacitated multi-source flows $\{f^r \,|\, r \in \mathcal{V}\}$ satisfying the constraints of Lemma 4. Then, from Lemma 4, it follows readily that every element of the RHS has to be an element of the polyhedron $P_{\mathbf{\Lambda}}$, which proves $P_{\mathbf{\Lambda}} \supseteq$ RHS of Equation (2.76).
- $P_{\mathbf{\Lambda}} \subseteq$ RHS of Equation (2.76): We have to show that, for every element $\boldsymbol{x}$ of the polyhedron $P_{\mathbf{\Lambda}}$, there exists a distribution $\boldsymbol{a}$ and a set of feasible multi-source flows $\{f^r \,|\, r \in \mathcal{V}\}$ satisfying the constraints of Lemma 4. Let $\boldsymbol{x}$ be an arbitrary element of the polyhedron $P_{\mathbf{\Lambda}}$, and let $\boldsymbol{\alpha}$ be a distribution over the set of spanning trees such that $\boldsymbol{\alpha}\mathbf{\Lambda} \leq \boldsymbol{x}$. Now, I show how a distribution $\boldsymbol{a}$ and a set of spanning trees $\{f^r \,|\, r \in \mathcal{V}\}$ satisfying the constraints of Lemma 4 can be found.

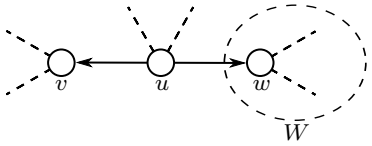


Figure 2.3: Illustration for the proof of Theorem 6.

First, for each spanning tree $T \in \mathcal{T}$, consider the orientation which directs each edge e such that it faces the non-smaller component of $G[T \setminus \{e\}]$.[18] I claim that the resulting directed spanning

[18] If the two components are of equal size, then choose an arbitrary direction.

tree is actually a reverse arborescence. We have to show that there is no pair of arcs $(u,v),(u,w)$: $u,v,w \in \mathcal{V}, v \neq w$ (see Figure 2.3 for an illustration). For the sake of contradiction, suppose that this is not true. Let W denote the node set of the larger component of $G[T \setminus \{(u,w)\}]$ (again, choose an arbitrary component if they are of equal size). Since W consists of the nodes of the larger component, we have $|W \cup \{u\}| \geq \frac{|\mathcal{V}|}{2} + 1 > \frac{|\mathcal{V}|}{2}$. But this leads to a contradiction as $W \cup \{u\}$ is a subset of the smaller component of $G[T \setminus \{(u,v)\}]$ due to the direction of (u,v). Thus, the claim that the resulting directed spanning tree is a reverse arborescence has to hold. Hence, for each spanning tree T, I can define the *center* of T to be the root of the reverse arborescence.

Now, let a_r be the sum weight of those spanning trees whose center is r. Formally, let $a_r = \sum_{T \in \mathcal{T} : \text{center of } T \text{ is } r} \alpha_T$. Then,

$$\sum_{r \in \mathcal{V}} a_r = \sum_{r \in \mathcal{V}} \sum_{T \in \mathcal{T} : \text{center of } T \text{ is } r} \alpha_T \tag{2.78}$$

$$= \sum_{T \in \mathcal{T}} \alpha_T = 1 \ . \tag{2.79}$$

Therefore, $\boldsymbol{\alpha}$ is indeed a distribution.

Next, for each r, let

$$f^r(u,v) = \sum_{T \in \mathcal{T} : \text{ center of } T \text{ is } r \wedge (u,v) \text{ is facing the center of } T} \alpha_T \lambda(T, \{u,v\}) \ , \tag{2.80}$$

where "(u,v) is facing the center of T" means that the direction of $\{u,v\}$ in the above orientation (where I defined the center of a spanning tree) is (u,v). First, I show that $\sum_{r \in \mathcal{V}} f^r(u,v) + f^r(v,u) \leq \sum_{T \in \mathcal{T}} \alpha_T \lambda(T, \{u,v\})$ for every edge $\{u,v\} \in \mathcal{E}$. Let $\{u,v\}$ be an arbitrary edge. Then,

$$\sum_{r \in \mathcal{V}} f^r(u,v) + f^r(v,u) \tag{2.81}$$

$$= \sum_{r \in \mathcal{V}} \sum_{T \in \mathcal{T} : \text{ center of } T \text{ is } r \wedge (u,v) \text{ is facing the center of } T} \alpha_T \lambda(T, \{u,v\})$$

$$+ \sum_{T \in \mathcal{T} : \text{ center of } T \text{ is } r \wedge (v,u) \text{ is facing the center of } T} \alpha_T \lambda(T, \{u,v\}) \tag{2.82}$$

$$= \sum_{r \in \mathcal{V}} \sum_{T \in \mathcal{T} : \text{ center of } T \text{ is } r} \alpha_T \lambda(T, \{u,v\}) \tag{2.83}$$

$$= \sum_{T \in \mathcal{T}} \alpha_T \lambda(T, \{u,v\}) \ . \tag{2.84}$$

It remains to show that, in flow f^r, each node $v \in \mathcal{V} \setminus \{r\}$ is a source producing an amount of a_r. Let $v \in \mathcal{V} \setminus \{r\}$ be an arbitrary node, let $T \in \mathcal{T}$ be an arbitrary spanning tree, and consider again the above orientation of T (where I defined the center of a spanning tree). Recall that, in this orientation, each edge e is facing the larger component of $G[T \setminus \{e\}]$. Thus, if we remove the outgoing edge of v, then all those nodes will be disconnected from the larger component which would be disconnected if we removed the incoming edges of v. Since we also disconnect v from the larger component when we remove its outgoing edge, we have that the sum of $\lambda(T,e)$ over every incoming edge e is equal to $\lambda(T, e_{\text{outgoing}}) - 1$, where e_{outgoing} is the outgoing edge. Furthermore, we have $\lambda(T,e) = 0$ for every $e \notin T$ by definition. Since this holds for every spanning tree, we have

that the net outgoing flow (i.e., the production) of node $v \in \mathcal{V} \setminus \{r\}$ in flow f^r is

$$\sum_{\{u,v\}\in\mathcal{E}} f^r(v,u) - f^r(u,v) \tag{2.85}$$

$$= \sum_{\{u,v\}\in\mathcal{E}} \sum_{T\in\mathcal{T}\colon \text{ center of } T \text{ is } r \wedge (v,u) \text{ is facing the center of } T} \alpha_T \lambda(T,\{u,v\})$$

$$- \sum_{T\in\mathcal{T}\colon \text{ center of } T \text{ is } r \wedge (u,v) \text{ is facing the center of } T} \alpha_T \lambda(T,\{u,v\}) \tag{2.86}$$

$$= \sum_{T\in\mathcal{T}\colon \text{ center of } T \text{ is } r} \alpha_T \Bigg(\sum_{\{v,u\}\in\mathcal{E}\colon (v,u) \text{ is facing the center of } T} \lambda(T,\{u,v\})$$

$$- \sum_{\{v,u\}\in\mathcal{E}\colon (u,v) \text{ is facing the center of } T} \lambda(T,\{u,v\}) \Bigg) \tag{2.87}$$

$$= \sum_{T\in\mathcal{T}\colon \text{ center of } T \text{ is } r} \alpha_T \left(\lambda(T, e_{\text{outgoing}}) - \sum_{\{v,u\}\in \text{ incoming edges of } v} \lambda(T,\{u,v\}) \right) \tag{2.88}$$

$$= \sum_{T\in\mathcal{T}\colon \text{ center of } T \text{ is } r} \alpha_T \cdot 1 \tag{2.89}$$

$$= a_r \ . \tag{2.90}$$

Finally, it is easy to see using a similar argument that the consumption of node r is $a_r(|\mathcal{V}|-1)$, as removing all of its incoming edges disconnects all $|\mathcal{V}|-1$ other nodes. Therefore, the distribution $\boldsymbol{a}$ and the set of feasible multi-source flows $\{f^r \,|\, r \in \mathcal{V}\}$ satisfy the constraints of Lemma 4.

To prove Equation (2.77), I use Lemma 1. More specifically, I first show how the characterization of Equation 2.76 can be mapped to Equation 2.9, and then use Lemma 1 to find a characterization of the blocker.

First, the distribution $\boldsymbol{a}$ and the set of flows $\{f^r \,|\, r \in \mathcal{V}\}$ can be represented by a vector $\boldsymbol{f} \in \mathbb{R}_{\geq 0}^{|\mathcal{V}|+2|\mathcal{V}||\mathcal{E}|}$ (one element for each node to represent the distribution and $2|\mathcal{E}|$ elements for each node to represent the set of $|\mathcal{V}|$ flows). Then, $\boldsymbol{S}$ is a matrix of size $|\mathcal{E}| \times (|\mathcal{V}| + 2|\mathcal{V}||\mathcal{E}|)$. The first $|\mathcal{E}|$ columns of $\boldsymbol{S}$ are all zero, since we do not compare $\boldsymbol{a}$ to $\boldsymbol{x}$. The following two columns are both $[1,0,0,\ldots,0]'$ (representing summing the two directions of the first flow along the first edge e_1 to be compared with x_{e_1}), the next two columns are both $[0,1,0,\ldots,0]'$ (representing summing the two directions of the first flow along the second edge e_2 to be compared with x_{e_2}), etc. Then, after $2|\mathcal{E}|$ columns, we repeat this pattern for the second flow, then for the third flow, etc.

Second, $\boldsymbol{C}$ is a matrix of size $(2+2|\mathcal{V}|(|\mathcal{V}|-1)) \times (|\mathcal{V}|+2|\mathcal{E}||\mathcal{V}|)$ and $\boldsymbol{c}$ is a vector of length $2+2|\mathcal{V}|(|\mathcal{V}|-1)$, as $1+|\mathcal{V}|(|\mathcal{V}|-1)$ equalities can be represented by $2+2|\mathcal{V}|(|\mathcal{V}|-1)$ inequalities. The first two rows of $\boldsymbol{C}$ and the first two elements of $\boldsymbol{c}$ correspond to $\boldsymbol{a}'\boldsymbol{1} = 1$. Hence, the first row of $\boldsymbol{C}$ consists of $|\mathcal{V}|$ ones and the remaining elements are all zero, and the first element of $\boldsymbol{c}$ is one; while the second row of $\boldsymbol{C}$ consists of $|\mathcal{V}|$ negative ones and the remaining elements are all zero, and the second element of $\boldsymbol{c}$ is negative one. The following $2|\mathcal{V}|(|\mathcal{V}|-1)$ rows represent the net outgoing flow equalities for all $|\mathcal{V}|(|\mathcal{V}|-1)$ pairs of flows and source nodes. Hence, the remaining elements of $\boldsymbol{c}$ are all zero. For the ith flow and the jth source node, the two rows are constructed as follows. In the first row, let the ith element be negative one and the remainder of the first $|\mathcal{V}|$ elements be all zero, while in the second row, let the ith element be one and the remainder of the first $|\mathcal{V}|$ elements be all zero. For a given directions of a given edge, let the corresponding element of the matrix be zero if the jth node is not an endpoint of the edge, let the element be one in the first row and negative one in the second row if the jth node is the source, and let the element be negative one in the first row and one in the second row if the jth node is the target.

Now, I show using Lemma 1 that the blocker is indeed characterized by Equation 2.34. The length of the vector $\boldsymbol{h}$ is $2+2|\mathcal{V}|(|\mathcal{V}|-1)$. For notational simplicity, let its first and second element be denoted by K^+ and K^-, let the first and second element corresponding to flow r and source node v be denoted $\pi_r^+(v)$ and $\pi_r^-(v)$, and let $\pi_r^+(r) \equiv \pi_r^-(r) \equiv 0$ for every $r \in \mathcal{V}$. Then, a vector $\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$ is an element of the blocker $bl(P_{\boldsymbol{\Lambda}})$ iff there exist $K^+, K^- \in \mathbb{R}_{\geq 0}$, $\boldsymbol{g} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$, and $\{\pi_r^+, \pi_r^- : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R}_{\geq 0} \mid r \in \mathcal{V}\}$

satisfying

$$\boldsymbol{g} \leq \boldsymbol{y} \tag{2.91}$$

$$\forall r \in \mathcal{V}: \quad \sum_{v \in \mathcal{V} \setminus \{r\}} \pi_r^+(v) - \pi_r^-(v) \geq K^+ - K^- \tag{2.92}$$

$$\forall r \in \mathcal{V}, e = \{u, v\} \in \mathcal{E}: \quad \pi_r^+(u) - \pi_r^-(u) - \pi_r^+(v) + \pi_r^-(v) \leq g_e \tag{2.93}$$

$$\forall r \in \mathcal{V}, e = \{u, v\} \in \mathcal{E}: \; -\pi_r^+(u) + \pi_r^-(u) + \pi_r^+(v) - \pi_r^-(v) \leq g_e \tag{2.94}$$

$$K^+ - K^- \geq 1 \; . \tag{2.95}$$

Now, observe that if a $\{\pi_r^+, \pi_r^- : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R}_{\geq 0} \mid r \in \mathcal{V}\}$ satisfies the above inequalities with some K^+, K^-, and $\boldsymbol{g}$, then it also does with $K^+ = 1$, $K^- = 0$, and $\boldsymbol{g} = \boldsymbol{y}$. Consequently, we have that a vector $\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$ is an element of the blocker $bl(P_{\boldsymbol{\Lambda}})$ iff there exist $\{\pi_r^+, \pi_r^- : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R}_{\geq 0} \mid r \in \mathcal{V}\}$ satisfying

$$\forall r \in \mathcal{V}: \quad \sum_{v \in \mathcal{V} \setminus \{r\}} \pi_r^+(v) - \pi_r^-(v) \geq 1 \tag{2.96}$$

$$\forall r \in \mathcal{V}, e = \{u, v\} \in \mathcal{E}: \quad \pi_r^+(u) - \pi_r^-(u) - \pi_r^+(v) + \pi_r^-(v) \leq y_e \tag{2.97}$$

$$\forall r \in \mathcal{V}, e = \{u, v\} \in \mathcal{E}: \; -\pi_r^+(u) + \pi_r^-(u) + \pi_r^+(v) - \pi_r^-(v) \leq y_e \; . \tag{2.98}$$

Furthermore, we can replace each $\pi_r^+(v) - \pi_r^-(v)$ with a $\pi_r(v) \in \mathbb{R}$ (notice that this variable is not constrained to be non-negative). Therefore, we have that a vector $\boldsymbol{y} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$ is an element of the blocker $bl(P_{\boldsymbol{\Lambda}})$ iff there exist $\{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$ satisfying

$$\forall r \in \mathcal{V}: \quad \sum_{v \in \mathcal{V} \setminus \{r\}} \pi_r(v) \geq 1 \tag{2.99}$$

$$\forall r \in \mathcal{V}, e = \{u, v\} \in \mathcal{E}: \quad \pi_r(u) - \pi_r(v) \leq y_e \tag{2.100}$$

$$\forall r \in \mathcal{V}, e = \{u, v\} \in \mathcal{E}: \; -\pi_r(u) + \pi_r(v) \leq y_e \; , \tag{2.101}$$

where $\pi_r(r) \equiv 0$ as before. Finally, the last two constraints are equivalent to

$$\forall r \in \mathcal{V}, e = \{u, v\} \in \mathcal{E}: \; |\pi_r(u) - \pi_r(v)| \leq y_e \; . \tag{2.102}$$

□

Combined with the results of Section 2.3.2, the above theorem can readily be used to compute the adversary's equilibrium payoff and an equilibrium adversarial strategy. Furthermore, similarly to the All-to-One model, we can also find an operator strategy forming an equilibrium with the adversary's strategy, since the algorithm from the proof of Lemma 4 can be used to compute a mixed operator strategy $\boldsymbol{\alpha}$ from an element $\boldsymbol{x}$ of the polyhedron $P_{\boldsymbol{\Lambda}}$ in polynomial-time. First, find a distribution $\boldsymbol{a}$ and a set of flows $\{f^r \mid r \in \mathcal{V}\}$ satisfying the constraints of Lemma 4. Second, find a mixed operator strategy $\boldsymbol{\alpha}$ using the algorithm presented in the proof of the lemma. Then, we have $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \leq x_e$ for every $e \in \mathcal{E}$; hence, $\boldsymbol{\alpha}$ is a mixed operator strategy for $\boldsymbol{x}$. Finally, the running time of the algorithm is clearly polynomial in the size of the input (i.e., the network) if we output only the non-zero coefficients.

2.5.3 Graph-Theoretic Metric

In the previous section, I have shown that the vulnerability $\theta_{max}(G)$ of a network in the All-to-One model is closely related to the *directed strength* $\pi(G)$ of the topology graph. This results links the graph-theoretic robustness of a network to game theory, which allows us to better understand network robustness and robustness metrics. The question naturally arises: can we link vulnerability to some elementary graph metric in the case of the All-to-All model with linear usage as well? In this subsection, I show that this is indeed possible by studying the relationship between the vulnerability $\theta_{max}(G)$ and the Cheeger constant $h(G)$ of a network.

Properties of the Extreme Points of the Blocker

I begin with a theorem that establishes important properties of the extreme points of the blocker $bl(P_{\mathbf{\Lambda}})$. These properties will allow us to formulate the problem $\max_{\boldsymbol{y} \in bl(P_{\mathbf{\Lambda}})} \theta(\boldsymbol{y})$ as a graph partitioning problem.

Theorem 7. *Let $\boldsymbol{\omega}$ be an extreme point of the blocker $bl(P_{\mathbf{\Lambda}})$, and let $\{V_1, \ldots, V_m\}$ be the node sets of the connected components resulting from the removal of the support of $\boldsymbol{\omega}$. Then, for any $\{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$ satisfying the RHS of Equation (2.77) with $\boldsymbol{\omega}$,*

1. *if nodes u and v are in the same component, then $\pi_r(u) = \pi_r(v)$ for every $r \in \mathcal{V}$,*
2. *if nodes r and v are in the same component, then $\pi_r(v) = 0$,*
3. *if $\{u, v\}$ is in the support of $\boldsymbol{\omega}$, then u and v are in different components,*[19]
4. *if edges e' and e'' both connect the same components (that is, if $e' = \{u', v'\}$ and $e'' = \{u'', v''\}$ such that $u', u'' \in V_i$ and $v', v'' \in V_j$), then $\omega_{e'} = \omega_{e''}$;*

and there exists a $\{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$ satisfying the RHS of Equation (2.77) with $\boldsymbol{\omega}$ such that,

5. *if nodes r and s are in the same component, then $\pi_r(v) = \pi_s(v)$ for every $v \in \mathcal{V}$.*

Proof. I prove each property separately.

1. If nodes u and v are connected by an edge e that is not in the support of $\boldsymbol{\omega}$, then we have $\pi_r(u) = \pi_r(v)$ readily, as $|\pi_r(u) - \pi_r(v)| \leq \omega_e = 0$ for every r. If nodes u and v are not connected by such an edge, then there has to be a path $\{u, p_1\}, \{p_1, p_2\}, \ldots, \{p_l, v\}$ in G consisting of only edges not in the support of $\boldsymbol{\omega}$, otherwise u and v would not be in the same component. Then, we have $\pi_r(u) = \pi_r(p_1) = \pi_r(p_2) = \ldots = \pi_r(p_l) = \pi_r(v)$ using the same argument.

2. First, we have $\pi_r(r) = 0$ by definition. Second, we have $\pi_r(v) = \pi_r(r)$ from the previous property. Thus, we have $\pi_r(v) = \pi_r(r) = 0$.

3. For the sake of contradiction, suppose that the claim does not hold for some edge $\{u, v\}$. In other words, suppose that there exists an edge $\{u, v\}$ that is in the support of $\boldsymbol{\omega}$ such that u and v are in the same component. From the first property, we have that $\pi_r(u) = \pi_r(v)$ for every $r \in \mathcal{V}$. Now, let $\boldsymbol{\omega}'$ be the following vector: $\omega'_e = \omega_e$ when $e \neq \{u, v\}$, and $\omega'_e = 0$ when $e = \{u, v\}$. Then, $\boldsymbol{\omega}'$ satisfies the RHS of Equation (2.77) with $\{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$, as $0 = |\pi_r(u) - \pi_r(v)| \leq \omega'_e = 0$ for every $r \in \mathcal{V}$. Thus, $\boldsymbol{\omega}'$ is an element of the blocker $bl(P_{\mathbf{\Lambda}})$. However, we also have $\boldsymbol{\omega}' \leq \boldsymbol{\omega}$ and $\omega'_e < \omega_e$, which leads to a contradiction with the assumption that $\boldsymbol{\omega}$ is an extreme point. Therefore, nodes u and v must be in different components.

4. For the sake of contradiction, suppose that the claim does not hold for some edges e' and e''. That is, suppose that there exist two edges $e' = \{u', v'\}$ and $e'' = \{u'', v''\}$ such that $u', u'' \in V_i$, $v', v'' \in V_j$, and $\omega_{e'} > \omega_{e''}$. From the first property, we have that $\pi_r(u') = \pi_r(u'')$ and $\pi_r(v') = \pi_r(v'')$ for every $r \in \mathcal{V}$. Therefore, $|\pi_r(v') - \pi_r(u')| = |\pi_r(v'') - \pi_r(u'')| \leq \omega_{e''}$ must hold for every $r \in \mathcal{V}$. Now, let $\boldsymbol{\omega}'$ be the following vector: $\omega'_e = \omega_e$ when $e \neq e'$, and $\omega'_e = \omega_{e''}$ when $e = e'$. Then, $\boldsymbol{\omega}'$ satisfies the RHS of Equation (2.77) with $\{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$, as $|\pi_r(v') - \pi_r(u')| \leq \omega_{e''} = \omega'_{e'}$ for every $r \in \mathcal{V}$. Thus, $\boldsymbol{\omega}'$ is an element of the blocker $bl(P_{\mathbf{\Lambda}})$. However, we also have $\boldsymbol{\omega}' \leq \boldsymbol{\omega}$ and $\omega'_{e'} < \omega_{e'}$, which leads to a contradiction with the assumption that $\boldsymbol{\omega}$ is an extreme point. Therefore, $\omega_{e'} = \omega_{e''}$ must hold.

5. I show that, given a set of potential functions $\{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$ for which the claim does not hold, one can construct a set of potential functions $\{\pi'_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$ for which the claim does hold. Let nodes r and s be a pair of nodes from the same component for which the claim does not hold, i.e., there exists a $v \in \mathcal{V}$ such that $\pi_r(v) \neq \pi_s(v)$. Now, let $\pi'_r = \pi_s$ and let $\pi'_u = \pi_u$ for every $u \neq r$. Since $\{\pi_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$ satisfies the RHS of Equation (2.77) with $\boldsymbol{\omega}$, so does $\{\pi'_r : \mathcal{V} \setminus \{r\} \mapsto \mathbb{R} \mid r \in \mathcal{V}\}$. By repeatedly applying the above step, we can construct a set of potential functions that satisfies the property in a finite number of steps.

[19] Note that this is equivalent to saying that, for an edge e whose endpoints are in the same component, $\omega_e = 0$ must hold.

□

Using these properties, we can express the vulnerability $\theta_{max}(G) = \max_{\boldsymbol{y} \in bl(P_{\boldsymbol{\Lambda}})} \theta(\boldsymbol{y})$ of the graph G as a graph-partitioning problem. More specifically, since the maximum is always attained at an extreme point of the blocker, we can restrict the search space using the above properties. Using the first, second, and fifth properties, we can replace the set of potential functions in Equation (2.77), which has a function for each node and whose domains are the set of nodes, with a limited set of potential functions, which has a function for each connected component and whose domains are the set of components. Furthermore, using the third and fourth lemmas, we can replace the summations over the set of edges with summations over the set of edges between components. Then, we can express the problem $\max_{\boldsymbol{y} \in bl(P_{\boldsymbol{\Lambda}})} \theta(\boldsymbol{y})$ as follows.

Corollary 1. *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph, and let $\boldsymbol{\Lambda}$ be the usage (or loss) matrix of the All-to-All model with linear usage. The vulnerability $\theta_{max}(G)$ of the graph G is the solution of the following problem:*

$$\textit{Maximize } \frac{1}{\sum_{i<j} |E(V_i, V_j)| \max_r |\pi_{V_r}(V_i) - \pi_{V_r}(V_j)|} \left(1 - \sum_{i<j} \sum_{e=\{u,v\} \in E(V_i, V_j)} \mu_e \max_r |\pi_{V_r}(V_i) - \pi_{V_r}(V_j)| \right) \tag{2.103}$$

subject to

$$\forall r : \sum_i |V_i| \pi_{V_r}(V_i) \geq 1 \tag{2.104}$$

$$\forall r : \pi_{V_r}(V_r) = 0 \ , \tag{2.105}$$

where $\{V_1, \ldots, V_m\}$ is a partitioning of the nodes, $\{\pi_{V_i} : \{V_1, \ldots, V_m\} \mapsto \mathbb{R} \mid V_i \in \{V_1, \ldots, V_m\}\}$ is a set of potential functions, and $E(V_i, V_j)$ denotes the set of edges between V_i and V_j.

Relation to the Cheeger Constant

In graph theory, the *Cheeger constant* [Chung, 1997, Chung, 2005] (also called the *edge expansion coefficient* [Alon, 1997, Alon, 1998] or the *isoperimetric number* [Mohar, 1989, Mohar, 1988]) of a graph is a measure of "bottleneckedness". This metric is related to the spectral (or eigenvalue) gap of graph by the Cheeger inequalities and also has interesting applications, such as spectral clustering [Bühler and Hein, 2009]. In this subsection, I show that the special case of the All-to-All network blocking game where the attack costs μ_e are all zero is closely related to the Cheeger constant. Formally, assume that $\boldsymbol{\mu} = \mathbf{0}$ for the remainder of this section.

The Cheeger constant of a graph is defined as follows.

Definition 4 (Cheeger constant)**.** The *Cheeger constant* of a graph G, denoted by $h(G)$, is

$$h(G) = \min \left\{ \frac{|E(U, \mathcal{V} \setminus U)|}{|U|} : U \subset \mathcal{V}, 0 < |U| \leq \frac{|\mathcal{V}|}{2} \right\} \ , \tag{2.106}$$

where $E(U, \mathcal{V} \setminus U)$ is the set of all edges between U and $\mathcal{V} \setminus U$.

If $h(G)$ is low, then there is a relatively small set of edges that partitions the graph into two connected components which are both relatively large. In other words, if $h(G)$ is low, then there is a set of edges forming a "bottleneck" of the graph. Intuitively, these bottlenecks are weaknesses in the graph, which should correspond to the support of an optimal attack against the network. We will see that this is indeed true for many graphs.[20] The following theorem establishes the general relationship between the robustness of a graph in the All-to-All model with linear loss and the Cheeger constant.

Theorem 8. *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph, assume that $\boldsymbol{\mu} = \mathbf{0}$, and let $\theta_{max}(G)$ be the vulnerability of G in the All-to-All model with linear usage. Then, for every graph G,*

$$\theta_{max}^{-1}(G) \leq h(G) \ . \tag{2.107}$$

[20] As a first example, note that this is true for the network shown in Figure 2.2.

The proof of the theorem shows that our robustness metric $\theta_{max}^{-1}(G)$ can be interpreted as a possible "generalization" of the Cheeger constant $h(G)$ to arbitrary partitionings. Note that our robustness metric is also more general in the sense that it can incorporate attack costs, while the Cheeger constant cannot.

Proof. I show that the inverse of the value of the optimization problem in Corollary 1 is upper bounded by $h(G)$. For $\boldsymbol{\mu} = \mathbf{0}$, the value of the optimization problem is

$$\max \frac{1}{\sum_{i<j} |E(V_i, V_j)| \max_r |\pi_{V_r}(V_i) - \pi_{V_r}(V_j)|} \tag{2.108}$$

subject to the constraints in Corollary 1, while the inverse of the optimization problem's value is

$$\min \sum_{i<j} |E(V_i, V_j)| \max_r |\pi_{V_r}(V_i) - \pi_{V_r}(V_j)| \tag{2.109}$$

subject to the constraints in Corollary 1.

Now, consider the restricted optimization problem where the search space is restricted to partitions consisting of two connected components, denoted by V_1 and V_2. Since we are restricting a minimization problem, the value of the restricted problem is an upper bound of the value of the original problem. For any partitioning, the optimal values of the potential functions are determined by the cardinalities of V_1 and V_2. More specifically, the only constraints that the potential functions have to satisfy are $\pi_{V_1}(V_2) \geq \frac{1}{|V_2|}$ and $\pi_{V_2}(V_1) \geq \frac{1}{|V_1|}$. Without loss of generality, let $|V_1| \leq |V_2|$. Then, the value of the restricted optimization problem is

$$\min_{V_1 \subset \mathcal{V}} |E(V_1, V_2)| \max\{\pi_{V_1}(V_2), \pi_{V_2}(V_1)\} \tag{2.110}$$

$$= \min_{V_1 \subset \mathcal{V}} |E(V_1, V_2)| \max\{\frac{1}{|V_2|}, \frac{1}{|V_1|}\} \tag{2.111}$$

$$= \min_{V_1 \subset \mathcal{V}} |E(V_1, V_2)| \frac{1}{|V_1|} \tag{2.112}$$

$$= \min_{V_1 \subset \mathcal{V}} \frac{|E(V_1, \mathcal{V} \setminus V_1)|}{|V_1|} \tag{2.113}$$

$$= h(G) \ . \tag{2.114}$$

Therefore, we have that

$$\theta_{max}^{-1}(G) \leq h(G) \ . \tag{2.115}$$

□

As opposed to the case of the All-to-One model and directed graph strength, we do not have a strict equality here. Hence, the question arises: how tight is the above bound? The following theorem shows that, even though the bound is not tight for every graph, there is an equality for an infinite number of graphs.

Theorem 9. *Let $G = (\mathcal{V}, \mathcal{E})$ be an undirected graph, assume that $\boldsymbol{\mu} = \mathbf{0}$, and let $\theta_{max}(G)$ be the vulnerability of G in the All-to-All model with linear usage. Then,*

- *there is a graph G such that $\theta_{max}^{-1}(G) < h(G)$,*
- *and there are an infinite number of graphs such that $\theta_{max}^{-1}(G) = h(G)$.*

Proof. I prove each case by providing an example (or a set of examples).

- Consider the complete graph K_3. It is easy to see that the Cheeger constant of K_3 is $h(K_3) = 2$. To prove the claim, I show that the adversary can achieve a higher payoff than $\frac{1}{h(K_3)} = \frac{1}{2}$ regardless of the strategy of the operator. Let the strategy of the adversary be $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$. In any pure strategy of the operator, two edges are used and the expected loss of both edges is 1, since one node is cut off by the removal of each edge; therefore, the expected payoff is $\frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 1 + \frac{1}{3} \cdot 0 = \frac{2}{3}$. Since this is true for every pure strategy of the operator, it is also true for every mixed strategy. Hence, the adversary's equilibrium payoff has to be higher than $\frac{1}{h(K_3)}$.

- Consider a complete graph K_{2n}, where $n \in \mathbb{Z}_+$. It is well-known that the Cheeger constant of a complete graph K_{2n} is $h(K_{2n}) = \lceil \frac{2n}{2} \rceil = n$. I prove $\theta_{max}^{-1}(K_{2n}) = h(K_{2n})$ by showing that $\theta_{max}(K_{2n}) \leq h^{-1}(K_{2n})$. I do so by describing an operator strategy that achieves $\frac{1}{n}$ expected payoff regardless of the strategy of the adversary. Let the strategy of the operator be the uniform distribution over the set consisting of every star subgraph S_{2n} of K_{2n}. There are $2n$ such stars; therefore, the probability of each star is $\frac{1}{2n}$. Each edge of the graph is contained by two stars, and the loss of an edge is 1 in both stars because one node is cut off by the removal of the edge in both stars. Thus, its expected loss is $2 \cdot \frac{1}{2n} \cdot 1 = \frac{1}{n}$. Since the expected loss every edge is $\frac{1}{n}$, so is the operator's expected payoff. Therefore, $\theta_{max}(K_{2n}) \leq \frac{1}{n} = h^{-1}(K_{2n})$ holds.

□

2.6 Budget Constraints

In the network blocking game models I have discussed so far, the operator is assumed to be interested solely in minimizing her expected losses resulting from deliberate attacks. In other words, the operator considers only security and disregards other economic and technical factors, such as her operating costs or the quality of the service she provides. In practice, however, network operators are not indifferent to such factors, and can consider them to be as important as – or more important than – security. Furthermore, as different strategy choices can lead to different operating costs or different quality of service, these factors have to be an integral part of the operator's decision making. For this reason, I generalize network blocking games in this section to include such economic or technical factors as costs for the operator.

In [Gueye and Marbukh, 2012], a usage-based cost model was introduced and discussed for the special case of the Supply-Demand communication model and – what I call here – the maximum cost constraint[21]. However, it did not provide results on the computational complexity of the constrained game. In this section, I first extend this cost model to network blocking games in general in Section 2.6.1, and introduce the maximum and the expected (or average) cost budget constraints in Section 2.6.2. Then, I provide results on the computational complexity of solving the maximum cost constrained game in Section 2.6.3 and the expected cost constrained game in Section 2.6.4.

2.6.1 Cost Model

Recall from Section 2.2 that $\lambda(T, e)$ was defined to be the usage of link e when the operator selects collection T. This usage can measure, for example, the amount of traffic that traverses the link or the number of active paths between pairs of network nodes that include the link. Now, assume that each link e has some *unit usage cost* w_e, so that the network operator incurs $w_e\lambda(T, e)$ cost for using link e when she selects collection T.

This abstract unit cost can model various economic and technical factors. The following list enumerates some application examples.

- *Direct financial costs:* If the operator does not fully own every network element in G, then she might have to pay a fee in order to use or lease some of the elements. In other words, the network G can model the ensemble of the operator's own nodes and links and those that are available to her only for some price. By letting w_e be the unit price of using link e, we can use the unit cost model to express the usage-based fee $\lambda(T, e)w_e$ of link e when the operator selects collection T.

- *Quality of service:* The choice of network elements can also affect the quality of the service provided by the operator, as different links can have different negative effects on the traffic that passes through them. For example, each network link can cause some delay or jitter to the traffic that traverses it. By letting the unit cost w_e of a link e be the delay (or jitter) on that link, we can use $\lambda(T, e)w_e$ to quantify the total amount delay (or jitter) experienced by all traffic passing through link e when collection T is used.

- *Random faults and reliability:* We can also use unit costs to model random faults in the network elements similarly to [Schwartz et al., 2011]. Suppose that link failures can be caused either by

[21] The maximum cost constraint is defined later, in Section 2.6.2.

the actions of the malicious adversary or by random events that are independent of the adversary. Assume that the occurrence of these two kinds of failures are mutually exclusive. Then, let the probability of the failure being caused by the actions of the adversary be ρ, and let the probability of the failure being caused by random events be $1-\rho$. Finally, assume that random faults happen according a fixed and known probability distribution $\boldsymbol{\gamma}$ (intuitively, γ_e is how unreliable link e is). Then, the operator's expected loss due to attacks is

$$\rho \sum_{T \in \mathcal{T}} \alpha_T \sum_{e \in E} \beta_e \lambda(T, e) \ , \tag{2.116}$$

and her expected loss due to random faults is

$$(1-\rho) \sum_{T \in \mathcal{T}} \alpha_T \sum_{e \in E} \gamma_e \lambda(T, e) \ . \tag{2.117}$$

If we divide her loss values by ρ, which does not affect her strategy choice, then we can model losses due to random faults using a unit cost of $w_e = \frac{(1-\rho)\gamma_e}{\rho}$.

- *Resource usage:* Using a network element can entail resource usage (for example, electric energy consumption), which in turn can require expenditure from the operator. In this case, we use the unit cost w_e to model the unit resource usage cost (for example, the unit cost of electric energy) of link e, and $\lambda(T, e)w_e$ is the total resource usage cost entailed by link e when the operator selects collection T.

For the remainder of this section, I will assume that these economic and technical factors have been added together for each link e, and a constant $\boldsymbol{w} \in \mathbb{R}_{\geq 0}^{|\mathcal{E}|}$ is given. Furthermore, I will refer to all of these factors together as *costs*, and call w_e the unit cost of link e.

Based on the above definition of the unit cost of links, I now define the cumulative cost of collection (i.e., the cost of a pure strategy) and the cost of a mixed strategy. In the examples discussed above, the costs of the links can generally be assumed to be additive. For example, if traffic traverses multiple links, then the total amount of delay experienced is the sum of the individual delays; if multiple network elements consume electric energy, then the total energy consumption is the sum of the individual consumptions. Hence, the cumulative cost incurred by the operator when she selects collection T is the sum of the cost of each link. Formally, I define the *cumulative cost of a collection* T as

$$w(T) := \sum_{e \in \mathcal{E}} \lambda(T, e) w_e \tag{2.118}$$

$$= \boldsymbol{\lambda}_T \boldsymbol{w} \ . \tag{2.119}$$

Then, based on the cumulative cost of each collection, we can compute the expected – or average – cumulative cost when collections are selected according to a mixed strategy. Formally, the *expected usage cost of a mixed strategy* $\boldsymbol{\alpha}$ is

$$w(\boldsymbol{\alpha}) := \sum_{T \in \mathcal{T}} \alpha_T w(T) \tag{2.120}$$

$$= \sum_{T \in \mathcal{T}} \alpha_T \sum_{e \in \mathcal{E}} \lambda(T, e) w_e \tag{2.121}$$

$$= \boldsymbol{\alpha} \boldsymbol{\Lambda} \boldsymbol{w} \ . \tag{2.122}$$

2.6.2 Budget Constraint Formulations

We can incorporate this cost model into the game in multiple ways. First, we can assume that the operator has a fixed *budget* $b \in \mathbb{R}_{\geq 0}$ to spend, and her goal is to minimize her expected loss by choosing the most secure strategy whose cost does not exceed her budget. This constraint on her strategy choice can again be formulated in multiple ways. In the following subsections, I introduce two straightforward formulations, called the maximum and the expected cost budget constraints. Note that, in both cases, I assume that the budget b is large enough so that there exists at least one strategy whose cost does

not exceed it. Finally, we can use the attacker's equilibrium payoff in the constrained game, denoted by $\theta_{max}(b)$, to quantify the vulnerability of the network for an operator having a fixed budget b.

Second, we can assume that the operator has a flexible budget, and her goal is to minimize the vulnerability of the network and her budget at the same time. In this case, the operator has to solve the trade-off problem

$$\min_b \upsilon\theta_{max}(b) + b \ , \tag{2.123}$$

where υ measures how valuable security is to the operator. Notice that, once we are able to solve the fixed-budget problem efficiently, we are also able to solve the trade-off problem efficiently using a simple binary search. Consequently, I will focus on the computational complexity of solving the game with a fixed budget.

Maximum Cost Budget Constraint

Under the *maximum cost constraint* (MCC), the operator can use only those feasible collections whose cumulative costs (see Equation (2.118)) are less than or equal to the budget b. Formally, the operator's pure-strategy set is restricted to

$$\mathcal{T}^{(b)} := \{T \in \mathcal{T} \mid w(T) \leq b\} \ . \tag{2.124}$$

This formulation is best-suited for situations where the budget limit is determined by the amount of preallocated resources available. For example, when the unit costs correspond to electric energy consumption values and the budget corresponds to the amount of electric power available, the cumulative cost of the selected collection should never exceed the budget.

Expected Cost Budget Constraint

For some situations, the maximum cost formulation can be considered too strict. For example, when the amount of allocated resources can be modified during operation (e.g., when resources can be leased), the budget has to be compared to the average or – equivalently – the expected cost incurred by the operator during continuous operation. To capture these situations, I introduce a second budget constraint formulation, called the expected cost constraint.

Under the *expected cost constraint*, the operator can use a mixed strategy only if its expected cost (see Equation (2.120)) is less than or equal to the budget b. Formally, the set of mixed strategies available to the operator is

$$\mathcal{A}^{(b)} := \left\{ \boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^{|\mathcal{T}|} \;\middle|\; w(\boldsymbol{\alpha}) \leq b \ \wedge \ \boldsymbol{\alpha}'\mathbf{1} = 1 \right\} \ . \tag{2.125}$$

Note that the pure-strategy set of the operator is not constrained in this formulation (i.e., it is the set of feasible collections as in the unconstrained game). Furthermore, notice that this formulation generalizes the classic notion of mixed strategies in game theory, where a mixed strategy can be any distribution over the set of pure strategies. Here, the operator chooses her mixed strategy from a predefined subset of the distributions.

2.6.3 Computational Complexity in the Maximum Cost Constrained Game

In this subsection, I study the computational complexity of solving network blocking games under the maximum cost constraint. Since we have from Theorem 2 that solving an unconstrained NBG is NP-hard in general, we readily have that solving an NBG under any budget constraint is also NP-hard in general, as the unconstrained game is the special case of $b \to \infty$.[22] For this reason, I focus on the computational complexity of those communication models for which efficient algorithms exist in the unconstrained game. More specifically, in this subsection, I show that solving a network blocking game in the All-to-One, the All-to-All with linear usage, and the Supply-Demand communication models is NP-hard under the maximum cost constraint.

I prove the hardness of solving network blocking games under the maximum cost constraint by reducing a well-known NP-hard problem, the *Partition Problem* (PP) [Mertens, 2006], to the problem of computing the adversary's equilibrium payoff in the constrained game. More specifically, I reduce PP

[22] More precisely, the unconstrained game is the special case of a budget $b \in \mathbb{R}_{\geq 0}$ that is large enough such that the operator can use any strategy.

to the problem of determining whether the adversary's equilibrium payoff in a given network under the maximum cost constraint is less than or equal to a given value, which I call the *Equilibrium Problem with Maximum Cost Constraint* (EPMAX). The decision versions of these two problems are defined as follows.

Definition 5 (Partition Problem [PP])**.** Given a multiset of positive integers $\{x_1, \ldots, x_n\}$, is there a partitioning of the multiset into two disjoint subsets A and B such that

$$\sum_{x \in A} x = \sum_{x \in B} x \ ? \tag{2.126}$$

Definition 6 (Equilibrium Problem with Maximum Cost Constraint [EPMAX])**.** Given a communication model, a network G, a budget limit b, and a payoff threshold p, is the adversary's equilibrium payoff less than or equal to p?

Note that the only difference between the Equilibrium Problem (see Definition 2) and the Equilibrium Problem with Maximum Cost Constraint is that the latter has the maximum cost constraint in addition. Hence, the same argument about the relationship between the complexity of solving the game and solving the EP is valid in the case of EPMAX as well. Thus, if EPMAX is NP-hard, so is solving the game under the maximum cost constraint.

The following theorem shows that the Equilibrium Problem with Maximum Cost Constraint is NP-hard.

Theorem 10. *The Partition Problem is polynomial-time reducible to the Equilibrium Problem with Maximum Cost Constraint in the (a) Supply-Demand, the (b) All-to-All, and the (c) All-to-One communication models.*

Proof. For each communication model, I show how an instance of EPMAX (that is, a network, a budget limit, and a payoff threshold) can be constructed in polynomial time from an instance of PP. I then show that the instance of EPMAX is true if and only if the instance of PP has a solution.

To simplify the notations in the proofs, I define the *expected loss* of an edge $e \in \mathcal{E}$, denoted by $L(e)$, in a given operator strategy $\boldsymbol{\alpha}$ as

$$L(e) = \sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e) \ . \tag{2.127}$$

From the definitions of the players' payoffs, it follows readily that, when the adversary targets edge e, the adversary's expected payoff is $L(e) - \mu_e$ and the operator's expedted loss is $L(e)$.

Proof of Theorem 10 for the Supply-Demand Communication Model

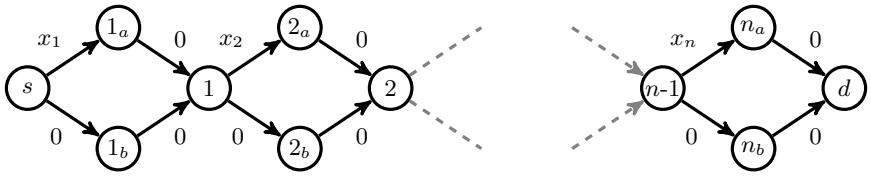


Figure 2.4: Illustration for the proof of Theorem 10 for the Supply-Demand model. Numbers along edges indicate unit costs.

Given an instance of PP, I build an instance of EPMAX as follows.

- Let the topology of the network be the following (see Figure 2.4 for an illustration): There is one source node, denoted by s, one sink node, denoted by d, and $3n-1$ other nodes, which are denoted by 1_a, 1_b, 1, 2_a, 2_b, 2, …, n_a, and n_b.

  Node s is connected to nodes 1_a and 1_b with edges having unit costs of x_1 and 0, respectively. Nodes i_a and i_b, for every $i < n$, are connected to node i with edges having zero unit cost. Node i is connected to nodes $(i+1)_a$ and $(i+1)_b$ with edges having unit costs of x_{i+1} and 0, respectively. Finally, nodes n_a and n_b are connected to node d with edges having zero unit cost.

- Let the capacity of the links and the amount of goods to be transported from s to d be 1.
- Let the operator's budget be $b = \frac{1}{2}\sum_{i=1}^{n} x_i$.
- Let the threshold payoff value be $p = \frac{1}{2}$.

I claim that the adversary's equilibrium payoff in the above network is greater than p if and only if PP does not have a solution.

First, suppose that the set can be partitioned into two subsets A and B of equal sum, in other words, suppose that PP has a solution. In this case, we have to show that the equilibrium payoff is at most $\frac{1}{2}$. First, notice that, since the total amount of goods to be transported from s to d is 1 and the amount of flow on each edge is either 0 or 1, the set of feasible integer flows is equal to the set of s-d paths in the graph, as the amount of flow on each edge is either 0 or 1.

Now, I show that there exist two disjoint paths (or, equivalently, flows) that satisfy the operator's budget constraint. The first path (or, equivalently, the first set of links with positive flow values) consists of the edges $(i-1, i_a)$ and (i_a, i) for each $x_i \in A$, and of the edges $(i-1, i_b)$ and (i_b, i) for each $x_i \notin A$. The second path consists of the remaining edges. In other words, the first flow takes the "path above" whenever $x_i \in A$ and the "path below" whenever $x_i \notin A$, while the second flow does the contrary. It is easy to see that the cost of both flows is $\sum_{x_i \in A} x_i = \sum_{x_i \in B} x_i = \frac{1}{2}\sum_i x_i$; thus, they both satisfy the maximum budget constraint. By assigning a probability of $\frac{1}{2}$ to each flow, we obtain an operator strategy in which the expected loss of every edge is at most $\frac{1}{2}$. If the operator employs this strategy, then the adversary's payoff for every pure and, consequently, every mixed strategy is at most $\frac{1}{2}$. Therefore, the equilibrium payoff cannot be greater than $\frac{1}{2}$. Hence, if PP has a solution, EPMAX has to be true.

Second, suppose that the set cannot be partitioned into two subsets of equal sum, that is, suppose that PP does not have a solution. In this case, we have to show that the adversary's equilibrium payoff is greater than $\frac{1}{2}$. If the equilibrium payoff were at most $\frac{1}{2}$, then there would exist an operator strategy $\boldsymbol{\alpha}$ in which the expected loss of every edge were at most $\frac{1}{2}$. I now show that no such strategy can exist.

Because of the maximum cost budget constraint, the cost of every pure operator strategy is less than or equal to $b = \frac{1}{2}\sum_i x_i$. Moreover, we can show that this inequality has to be strict. As every pure strategy is an s-d path, if the cost of a pure strategy were equal to b, there would exist a subset of links $I \subsetneq \{1, 2, \ldots, n\}$ such that $\sum_{i \in I} x_i = b$. By letting $A = \{x_i \mid i \in I\}$ and $B = \{x_i \mid i \notin I\}$, we would get a solution for PP, which would contradict the supposition that the set cannot be partitioned. Thus, the cost of every pure strategy is strictly less than b and, as a consequence, the expected cost of every mixed strategy is also strictly less than b. Formally, we have

$$\sum_{e \in \mathcal{E}} L(e) w_e < b = \frac{1}{2}\sum_{i=1}^{n} x_i = \sum_{e \in \mathcal{E}} \frac{1}{2} w_e \ . \tag{2.128}$$

Now, observe that the expected loss $L(e)$ of an edge e in the S-D model is equal to the expected amount of flow on that edge. Since the total amount of goods to be transported is 1 and each pair of "above" and "below" edges (e.g., e_a and e_b) is an s-d cut, the sum of the flows on any pair of "above" and "below" edges is at least 1. Thus, for every pair of above and below edges e_a and e_b, we have $L(e_a) + L(e_b) \geq 1 = \frac{1}{2} + \frac{1}{2}$. By combining this with the initial supposition that the expected loss of every edge is at most $\frac{1}{2}$, we have that

$$\forall e \in \mathcal{E} : \ L(e) = \frac{1}{2} \tag{2.129}$$

and

$$\sum_{e \in \mathcal{E}} L(e) w_e = \sum_{e \in \mathcal{E}} \frac{1}{2} w_e \ . \tag{2.130}$$

However, this leads to a contradiction with Equation (2.128), which proves that no operator strategy can exist in which the expected loss of every edge is at most $\frac{1}{2}$. Therefore, if PP does not have a solution, then the equilibrium payoff has to be greater than $\frac{1}{2}$ and EPMAX is not true, which concludes the proof for this communication model.

Proof of Theorem 10 for the All-to-All Communication Model

For the All-to-All communication model, I construct an instance of EPMAX from an instance of PP as follows.

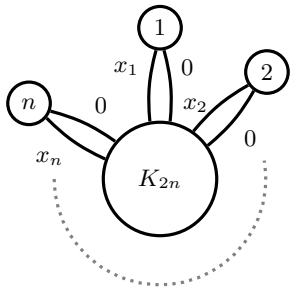


Figure 2.5: Illustration for the proof of Theorem 10 for the All-to-All model.

- Let the network topology be the following (see Figure 2.5 for an illustration): There is a large clique, which consists of $2n$ nodes, and there are n "outer" nodes, to which I refer as node 1, node 2, ..., node n. Each node i, for $i = 1, \ldots, n$, is connected to two distinct nodes of the clique with edges having unit costs of x_i and 0, such that every node in the clique is connected to exactly one outer node. Finally, edges between two nodes in the clique have zero unit cost.

- Let the operator's budget be $b = \frac{1}{2}\sum_{i=1}^{n} x_i$.

- Let the threshold payoff value be $p = \frac{1}{2}$.

I claim that the equilibrium payoff in the above network is greater than $\frac{1}{2}$ if and only if PP does not have a solution.

As in the previous proof, first suppose that PP has a solution (A, B). Then, I use this solution to derive an operator strategy in which the expected loss of every edge is at most $\frac{1}{2}$. However, instead of describing this mixed strategy explicitly, I give a randomized algorithm for selecting pure strategies, and use the distribution of this algorithm's output as the mixed strategy. The algorithm for selecting pure strategies (i.e., spanning trees) is the following. First, choose either set A or set B with equal probability (that is, choose them at random with probabilities $\frac{1}{2}, \frac{1}{2}$). Second, connect each outer node i to the clique with exactly one edge: if x_i belongs to the chosen set, then use the edge which has cost x_i; otherwise, use the other edge. Finally, complete the spanning tree by choosing a star subgraph of the clique uniformly at random.

Now, I show that the expected loss of every edge is at most $\frac{1}{2}$ if the operator uses this randomized algorithm as her mixed strategy. First, each outer edge e is used with probability $\frac{1}{2}$, and its removal cuts off at most one node; thus, we have $L(e) \leq \frac{1}{2}$ for the outer links. Second, each link e inside the clique is used with probability $\frac{1}{n}$ (the probability that a randomly chosen star subgraph contains it), and its removal cuts off at most two nodes; thus, we have $L(e) \leq \frac{2}{n}$ for the inner links.[23] Therefore, if PP has a solution, then EPMAX is true.

Next, suppose that PP does not have a solution. Then, we can use the same argument as before to show that the cost of every pure strategy and, hence, the expected cost of every mixed strategy is strictly less than b. That is,

$$\sum_{e \in E_{\text{outer}}} w_e L(e) < b = \frac{1}{2}\sum_{i} x_i = \sum_{e \in E_{\text{outer}}} \frac{1}{2} w_e \ , \tag{2.131}$$

where E_{outer} is the set of outer links. Now, consider an arbitrary pair of edges e_a and e_b which connect an outer node to the clique. It can be shown easily that $L(e_a) + L(e_b) \geq 1$. If there were an operator strategy in which the expected loss of every edge were at most $\frac{1}{2}$, then it would follow that $\forall e \in E_{\text{outer}} : \ L(e) = \frac{1}{2}$. However, this would lead to a contradiction with Equation (2.131); thus, no such strategy can exist. Therefore, if PP does not have a solution, then EPMAX is not true.

Proof of Theorem 10 for the All-to-One Communication Model

For the All-to-One communication model, I construct an instance of EPMAX from an instance of PP as follows.

[23] Note that we can assume $n \geq 4$ for the reduction.

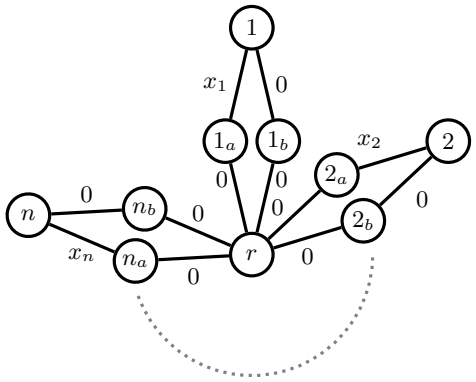


Figure 2.6: Illustration for the proof of Theorem 10 for the All-to-One model.

- Let the network topology be the following (see Figure 2.6 for an illustration): There is a designated node r, which is connected to $2n$ nodes (denoted by 1_a, 1_b, 2_a, 2_b, ..., n_a and n_b) in the form of a large star rooted at r. Furthermore, there are n "outer" nodes, denoted by 1, 2, ..., n. Node i is connected to nodes i_a and i_b with edges having unit costs of x_i and 0. The edges connecting nodes i_a and i_b to r both have zero unit cost.
- Let the operator's budget be $b = \frac{1}{2}\sum_{i=1}^{n} x_i$.
- Let the threshold payoff value be $p = \frac{3}{2}$.

I claim that the adversary's equilibrium payoff in the above network is greater than $\frac{3}{2}$ iff PP does not have a solution.

First, suppose that PP has a solution (A, B). Then, we can use this solution to derive an operator strategy in which the expected loss of every edge is at most $\frac{3}{2}$. Again, instead of describing this mixed strategy explicitly, I give a randomized algorithm for selecting pure strategies, and use the distribution of this algorithm's output as the mixed strategy. The algorithm for selecting pure strategies (i.e., spanning trees) is the following. First, choose either A or B with equal probability. Second, connect each outer node i to the clique as follows: if x_i belongs to the chosen set, then use the edge which has cost x_i; otherwise, use the other edge. Finally, always use the edges that are connected to r (i.e., use them with a probability of 1).

Now, I show that the expected loss of every edge is at most $\frac{3}{2}$. First, each outer edge e is used with probability $\frac{1}{2}$, and its removal cuts off at most 1 node; thus, we have $L(e) \leq \frac{1}{2}$ for the outer edges. Second, each inner edge e is used with probability 1, and the number of nodes cut off by its removal is 1 node with probability $\frac{1}{2}$ and 2 with probability $\frac{1}{2}$; thus, we have $L(e) \leq \frac{3}{2}$ for the outer edges. Therefore, we have $L(e) \leq \frac{3}{2}$ for every edge e, which proves that EPMAX is true if PP has a solution.

Second, suppose that PP does not have a solution. Then, we have that the cost of every pure strategy is strictly less than b, which implies

$$\sum_{e \in E_{\text{outer}}} w_e L(e) < b = \frac{1}{2}\sum_{i} x_i = \sum_{e \in E_{\text{outer}}} \frac{1}{2} w_e \ , \tag{2.132}$$

where E_{outer} is the set of outer edges. For a pair of edges $e_a = (i_a, r)$ and $e_b = (i_b, r)$, we can easily show that $L(e_a) + L(e_b) \geq 3$. If there were an operator strategy in which the expected loss of every edge was at most $\frac{3}{2}$, it would follow that $L(e_a) = L(e_b) = \frac{3}{2}$, which would imply that expected loss of every outer edge is $\frac{1}{2}$. However, this would lead to a contradiction with Equation (2.132); thus, no such strategy can exist. Therefore, if PP does not have a solution, then EPMAX is not true. □

2.6.4 Computational Complexity in the Expected Cost Constrained Game

In this subsection, I study the computational complexity of solving network blocking games under the expected cost constraint. Similarly to the previous section, I focus on those communication models for which efficient algorithms exist in the unconstrained game, as we already have from Theorem 2 that the

problem is generally NP-hard. However, instead of providing specific algorithms for particular communication models, I focus on the class of models for which the polyhedron $P_{\mathbf{\Lambda}}$ (of the unconstrained game) has a polynomial-size linear characterization (see Section 2.3.2). More specifically, in this subsection, I show how a network blocking game under the expected cost constraint can be solved in polynomial-time if the polyhedron $P_{\mathbf{\Lambda}}$ of the unconstrained game has a polynomial-size linear characterization.

First, note that we cannot apply Theorem 1 directly to an expected cost constrained game, since we have an extra constraint on the operator's mixed strategies, which the theorem does not consider. Consequently, in order to be able to use Theorem 1, for each constrained game, I define an unconstrained blocking game that is equivalent to the constrained game in certain respects. Most importantly, for a given constrained game, I define this *equivalent game* so that the adversary's mixed-strategy equilibrium payoff and her equilibrium strategies are the same as in the original, constrained game. This can be achieved easily, for example, by defining the operator's pure-strategy set in the equivalent game to be the set $\{\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^{|\mathcal{T}|} \mid \boldsymbol{\alpha}'\mathbf{1} = 1 \ \wedge \ \boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{w} \leq b\}$ (intuitively, the set of of all convex linear combination coefficients $\boldsymbol{\alpha}$ satisfying $\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{w} \leq b$), the operator's loss for a pure-strategy profile $(\boldsymbol{\alpha}, e)$ to be $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e)$, and the adversary's payoff to be the operator's loss minus μ_e. It is fairly easy to see that the adversary's mixed-strategy equilibrium payoff in the equivalent game is the same as in the original, constrained game.

However, the operator's pure-strategy set in the above equivalent game has the cardinality of the continuum, while Theorem 1 considers a finite set (i.e., a finite set of feasible collections). Consequently, we cannot use the set of all convex linear combination coefficients directly as the operator's pure-strategy set. I sidestep this problem by using only the extreme points of this set. The following definition formalizes the equivalent unconstrained game.

Definition 7 (Equivalent Unconstrained Game)**.** Let $\mathbf{\Lambda}$ be the usage (or loss) matrix of a blocking game, let $\mathcal{T}$ and $\mathcal{E}$ be the operator's and the adversary's pure-strategy sets respectively, let $\boldsymbol{\mu}$ and $\boldsymbol{w}$ be the attack and usage costs respectively, and let $b \in \mathbb{R}_{\geq 0}$ be a budget value.

Then, the *equivalent unconstrained game* is defined as follows. Let the operator's pure-strategies be the extreme points of the set $\{\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^{|\mathcal{T}|} \mid \boldsymbol{\alpha}'\mathbf{1} = 1 \ \wedge \ \boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{w} \leq b\}$, let the adversary's pure-strategy set be $\mathcal{E}$, and for a pure-strategy profile $(\boldsymbol{\alpha}, e)$, let the operator's loss be $\sum_{T \in \mathcal{T}} \alpha_T \lambda(T, e)$ and the adversary's payoff be the operator's loss minus μ_e.

It is easy to see that the operator's pure-strategy set in the equivalent game is finite. First, observe that the set $\{\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^{|\mathcal{T}|} \mid \boldsymbol{\alpha}'\mathbf{1} = 1\}$ is a convex $(|\mathcal{T}| - 1)$-dimensional polyhedron with $|\mathcal{T}|$ extreme points. By adding the linear constraint $\boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{w} \leq b$, we cut the polyhedron with a hyperplane, and the resulting set is again a convex (at most) $(|\mathcal{T}|-1)$-dimensional polyhedron with a finite number of extreme points. Therefore, we can apply Theorem 1 to the equivalent unconstrained game. The following theorem shows that the equivalent game is indeed equivalent to the original game with respect to the adversary's equilibrium payoff and her set of equilibrium strategies. Consequently, it suffices to find an efficient algorithm only for solving the equivalent game.

Theorem 11. *For any blocking game, the adversary's equilibrium payoff and her set of equilibrium strategies*[24] *are the same in the original game under the expected cost constraint and in the equivalent unconstrained game.*

Proof. I first show that, for every mixed operator strategy in the original game, there exists a mixed operator strategy in the equivalent game such that the players' payoffs are the same in the two games for every adversarial strategy, and vice versa.

Let $\boldsymbol{\alpha}^*$ be a mixed operator strategy in the original game. Since $\boldsymbol{\alpha}^*$ has to satisfy the budget constraint, we have $\boldsymbol{\alpha}^*\mathbf{\Lambda}\boldsymbol{w} \leq b$. Consequently, $\boldsymbol{\alpha}^*$ is an element of $\{\boldsymbol{\alpha} \in \mathbb{R}_{\geq 0}^{|\mathcal{T}|} \mid \boldsymbol{\alpha}'\mathbf{1} = 1 \ \wedge \ \boldsymbol{\alpha}\mathbf{\Lambda}\boldsymbol{w} \leq b\}$. Since this set is convex, we have that $\boldsymbol{\alpha}^*$ can be expressed as a convex linear combination of the extreme points. Let $\boldsymbol{a}$ be the coefficients of this convex linear combination, that is, let $\boldsymbol{a}$ be such that $\sum_{\boldsymbol{\alpha} \in \text{ extreme points}} a_{\boldsymbol{\alpha}} \boldsymbol{\alpha} = \boldsymbol{\alpha}^*$. I claim that, for any $e \in \mathcal{E}$, the players' payoffs are the same in the original game with $\boldsymbol{\alpha}^*$ as the operator's strategy and in the equivalent game with $\boldsymbol{a}$. Since the adversary's attack costs are the same by definition, it suffices to show that the operator's loss is the same. In the original

[24] A mixed adversarial strategy is an equilibrium strategy if there exists a mixed operator strategy such that the two form an equilibrium.

game, the operator's loss is $\sum_{T\in\mathcal{T}} \alpha_T^* \lambda(T,e)$. In the equivalent game the operator's loss is

$$\sum_{\boldsymbol{\alpha}\in \text{ extreme points}} a_{\boldsymbol{\alpha}} \sum_{T\in\mathcal{T}} \alpha_T \lambda(T,e) \tag{2.133}$$

$$= \sum_{T\in\mathcal{T}} \lambda(T,e) \sum_{\boldsymbol{\alpha}\in \text{ extreme points}} a_{\boldsymbol{\alpha}} \alpha_T \tag{2.134}$$

$$= \sum_{T\in\mathcal{T}} \lambda(T,e) \alpha_T^* \ . \tag{2.135}$$

Thus, for any adversarial strategy, the players' payoffs have to be the same in the original game with $\boldsymbol{\alpha}^*$ as the operator's strategy and in the equivalent game with $\boldsymbol{a}$.

The other direction – that is, showing that there exists a corresponding mixed-strategy $\boldsymbol{\alpha}^*$ in the original game for every mixed-strategy $\boldsymbol{a}$ in the equivalent game – can be shown using a similar argument.

Now, I show that a mixed-adversarial strategy is an equilibrium strategy in the original game if and only if it is an equilibrium strategy in the equivalent game.

First, assume that $\boldsymbol{\beta}$ is an equilibrium strategy in the original game. In other words, assume that there exists a mixed operator strategy $\boldsymbol{\alpha}$ such that $(\boldsymbol{\alpha},\boldsymbol{\beta})$ is an equilibrium. Let $\boldsymbol{a}$ be the corresponding operator strategy in the equivalent game (see above). Then, I have to show that $\boldsymbol{a}$ and $\boldsymbol{\beta}$ are both best responses. First, suppose for the sake of contradiction that $\boldsymbol{a}$ is not a best response, that is, suppose that there exists a mixed-strategy $\tilde{\boldsymbol{a}}$ which yields smaller loss to the operator. Then, there exists a mixed-strategy $\tilde{\boldsymbol{\alpha}}$ in the original game corresponding to $\tilde{\boldsymbol{a}}$, which yields smaller loss to the operator than $\boldsymbol{\alpha}$. However, this leads to a contradiction with the assumption that $(\boldsymbol{\alpha},\boldsymbol{\beta})$ is an equilibrium; hence, $\boldsymbol{a}$ has to be a best response. Second, suppose for the sake of contradiction that $\boldsymbol{\beta}$ is not a best response to $\boldsymbol{a}$, that is, suppose there exists a mixed-strategy $\tilde{\boldsymbol{\beta}}$ which yields smaller loss to the operator. However, this again leads to a contradiction, as $\tilde{\boldsymbol{\beta}}$ being a better response to $\boldsymbol{a}$ implies that $\tilde{\boldsymbol{\beta}}$ is a better response to $\boldsymbol{\alpha}$ than $\boldsymbol{\beta}$. Thus, if $\boldsymbol{\beta}$ is an equilibrium strategy in the original game, then it also has to be an equilibrium strategy in the equivalent game.

Second, assume that $\boldsymbol{\beta}$ is an equilibrium strategy in the equivalent game. In other words, assume that there exists a mixed operator strategy $\boldsymbol{a}$ such that $(\boldsymbol{a},\boldsymbol{\beta})$ is an equilibrium. Then, using similar arguments as in the preceding paragraph, we can show that the mixed-strategy $\boldsymbol{\alpha}$ in the original game corresponding to $\boldsymbol{a}$ forms an equilibrium with $\boldsymbol{\beta}$. Therefore, the set of the adversary's equilibrium strategies is the same in the original game and in the equivalent game.

Finally, I show that the adversary's equilibrium payoff is equal in the two games. Note that, at this point, we do not have that the adversary's payoff is the same in every equilibrium of the original game, as Theorem 1 cannot be applied to this game. However, we will soon see that the adversary's equilibrium payoff has to be unique in the original game. First, I show that, for every mixed adversarial strategy $\boldsymbol{\beta}$, the adversary's payoff is the same in the two games given that the operator plays a best response. By definition, the adversary's payoff is the loss of the operator minus $\boldsymbol{\beta}'\boldsymbol{\mu}$ in both games. Hence, it suffices to show that the operator's loss is the same given that she plays a best response. For the sake of contradiction, first suppose that the operator's loss in the original game for a best response $\boldsymbol{\alpha}$ is lower than her loss in the equivalent game for a best response $\boldsymbol{a}$. However, this leads to a contradiction, as the strategy $\tilde{\boldsymbol{a}}$ in the equivalent game corresponding to $\boldsymbol{\alpha}$ is a better response to $\boldsymbol{\beta}$ than $\boldsymbol{a}$. By using the same argument, we can show that the operator's loss cannot be lower in the equivalent game than in the original. Hence, the adversary' payoff is the same in the two games for any $\boldsymbol{\beta}$ given that the operator plays a best response. Since the operator plays a best response in an equilibrium, we have that the players' payoffs are the same for any equilibrium strategy profile. Finally, by combining this with the uniqueness of the adversary's payoff in the equivalent game, we have that the adversary has the same unique payoff for all the equilibria of the two games. □

It remains to show that the equivalent game can be solved efficiently. To prove this, I build on the results of Section 2.3.2 and show that the polyhedron of the equivalent game has a polynomial-size linear characterization (given that the polyhedron of the original game has one). Then, it follows readily that the equivalent game can be solved efficiently.

Lemma 5. *Let $\boldsymbol{\Lambda}$ be the usage (or loss) matrix of a blocking game, let $\boldsymbol{w}$ be the usage costs of the edges, let $b \in \mathbb{R}_{\geq 0}$ be a budget value, and assume that the polyhedron $P_{\boldsymbol{\Lambda}}$ has a polynomial-size linear*

characterization

$$P_{\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \;\wedge\; \boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \right) \right\} . \tag{2.136}$$

Then, the polyhedron of the equivalent unconstrained game can be characterized as

$$\left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \;\wedge\; \boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{S}\boldsymbol{f}\boldsymbol{w} \leq b \right) \right\} . \tag{2.137}$$

Observe that the two characterizations are identical except for the additional constraint $\boldsymbol{S}\boldsymbol{f}\boldsymbol{w} \leq b$ in the case of the equivalent game. Hence, to solve a blocking game under the expected cost constraint with usage costs $\boldsymbol{w}$ and budget b, one has to simply use the polynomial-size characterization of the unconstrained polyhedron $P_{\boldsymbol{\Lambda}}$ with the additional constraint $\boldsymbol{S}\boldsymbol{f}\boldsymbol{w} \leq b$. In other words, the equivalent game can be "skipped" in practice, and the game can be solved directly using Equation (2.137) as the polyhedron of the game and the results of Section 2.3.2.

Proof. Let $\boldsymbol{A}$ be the matrix whose rows are the extreme points of $\{\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \mid \boldsymbol{\alpha}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda}\boldsymbol{w} \leq b\}$. Then, the loss (or usage) matrix of the equivalent game is $\boldsymbol{A}\boldsymbol{\Lambda}$ (see Definition 7). By definition, the polyhedron $P_{\boldsymbol{A}\boldsymbol{\Lambda}}$ is

$$P_{\boldsymbol{A}\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{a} \in \mathbb{R}^{n}_{\geq 0} \left(\boldsymbol{a}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{a}\boldsymbol{A}\boldsymbol{\Lambda} \leq \boldsymbol{x} \right) \right\} , \tag{2.138}$$

where n is the number of rows of $\boldsymbol{A}$. Now, since the rows of $\boldsymbol{A}$ are the extreme points of the convex polyhedron $\{\boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \mid \boldsymbol{\alpha}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda}\boldsymbol{w} \leq b\}$, we can express $P_{\boldsymbol{A}\boldsymbol{\Lambda}}$ as

$$P_{\boldsymbol{A}\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \left(\boldsymbol{\alpha}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda}\boldsymbol{w} \leq b \right) \right\} . \tag{2.139}$$

By introducing a "dummy" variable $\boldsymbol{z}$, we can reformulate the above expression as

$$\begin{aligned}
P_{\boldsymbol{A}\boldsymbol{\Lambda}} &= \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{z} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}, \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \left(\boldsymbol{\alpha}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{z} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda} = \boldsymbol{z} \;\wedge\; \boldsymbol{z}\boldsymbol{w} \leq b \right) \right\} && (2.140) \\
&= \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{z} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \left(\boldsymbol{z} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{z}\boldsymbol{w} \leq b \;\wedge\; \exists \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \left(\boldsymbol{\alpha}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda} = \boldsymbol{z} \right) \right) \right\} && (2.141) \\
&= \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{z} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \left(\boldsymbol{z} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{z}\boldsymbol{w} \leq b \;\wedge\; \exists \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \left(\boldsymbol{\alpha}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda} \leq \boldsymbol{z} \right) \right) \right\} . && (2.142)
\end{aligned}$$

Note that the last step is correct because, if there exists a $(\boldsymbol{z}, \boldsymbol{\alpha})$ satisfying the RHS with the constraint $\boldsymbol{\alpha}\boldsymbol{\Lambda} \leq \boldsymbol{z}$, then $(\boldsymbol{\alpha}\boldsymbol{\Lambda}, \boldsymbol{\alpha})$ satisfies the RHS with the constraint $\boldsymbol{\alpha}\boldsymbol{\Lambda} = \boldsymbol{z}$.

Now, I use the polynomial-size linear characterization of the original polyhedron $P_{\boldsymbol{\Lambda}}$. Recall that this characterization can be expressed as

$$\forall \boldsymbol{z} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} : \; \exists \boldsymbol{\alpha} \in \mathbb{R}^{|\mathcal{T}|}_{\geq 0} \left(\boldsymbol{\alpha}'\boldsymbol{1} = 1 \;\wedge\; \boldsymbol{\alpha}\boldsymbol{\Lambda} \leq \boldsymbol{z} \right) \Longleftrightarrow \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \;\wedge\; \boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{z} \right) . \tag{2.143}$$

Consequently, we have

$$\begin{aligned}
P_{\boldsymbol{A}\boldsymbol{\Lambda}} &= \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{z} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \left(\boldsymbol{z} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{z}\boldsymbol{w} \leq b \;\wedge\; \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \;\wedge\; \boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{z} \right) \right) \right\} && (2.144) \\
&= \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{z} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \left(\boldsymbol{z} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{z}\boldsymbol{w} \leq b \;\wedge\; \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \;\wedge\; \boldsymbol{S}\boldsymbol{f} = \boldsymbol{z} \right) \right) \right\} && (2.145) \\
&= \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{z} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0}, \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{z} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{z}\boldsymbol{w} \leq b \;\wedge\; \boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \;\wedge\; \boldsymbol{S}\boldsymbol{f} = \boldsymbol{z} \right) \right\} . && (2.146)
\end{aligned}$$

Finally, by removing the dummy variable $\boldsymbol{z}$, we can characterize the blocker of the equivalent game as

$$P_{\boldsymbol{A}\boldsymbol{\Lambda}} = \left\{ \boldsymbol{x} \in \mathbb{R}^{|\mathcal{E}|}_{\geq 0} \;\middle|\; \exists \boldsymbol{f} \in \mathbb{R}^{k}_{\geq 0} \left(\boldsymbol{C}\boldsymbol{f} \geq \boldsymbol{c} \;\wedge\; \boldsymbol{S}\boldsymbol{f} \leq \boldsymbol{x} \;\wedge\; \boldsymbol{S}\boldsymbol{f}\boldsymbol{w} \leq b \right) \right\} . \tag{2.147}$$

□

2.7 Related Work on the Robustness of Network Topologies

In this section, I discuss some of the previously proposed graph metrics that can be used to quantify robustness. I begin with more basic, graph-theoretic notions, such as connectivity, and then give examples of studies that are based on assuming more strategic adversaries.

2.7.1 Graph-Theoretic Metrics

Connectivity The vertex-connectivity (or edge-connectivity) of a graph measures the minimum number of vertices (or edges) that have to be removed in order to disconnect the graph. As a more formal definition, let us say that a graph is k-vertex-connected (or k-edge-connected) if it remains connected whenever fewer than k vertices (or edges) are removed, and let the vertex-connectivity (or edge-connectivity) of a graph be the largest k for which it is k-vertex-connected (or k-edge-connected). Connectivity has many appealing theoretical-properties, such as being closely related to the number of independent paths between vertices (see Menger's theorem). Furthermore, the problem of computing the connectivity of a graph can be solved in polynomial time.

These metrics are very widely used for measuring the robustness of network topologies in the literature. For example, in the case of wireless sensor networks, vertex- and edge-connectivity are undoubtedly the most prevalent metrics [Li et al., 2009, Younis and Akkaya, 2008, Misra et al., 2008, Kashyap et al., 2006, Zhang et al., 2007, Han et al., 2007]. For another example in support of using connectivity, see [Dekker and Colbert, 2004].

Unfortunately, connectivity as a measure of topology robustness against strategic attacks has some weaknesses, which limit its practical usage. Firstly, connectivity is only concerned with the size of smallest disconnecting attack. In practice, however, the maximum attack-size that should be anticipated – in terms of the number of vertices (or edges) that the adversary can remove from the network – may be difficult to estimate. Secondly, connectivity is only concerned with whether an attack disconnects a network or not. In other words, connectivity does not take into account how disintegrated the network becomes as a result of an attack. In practice, however, an operator needs to care about how much functionality is retained by the network.

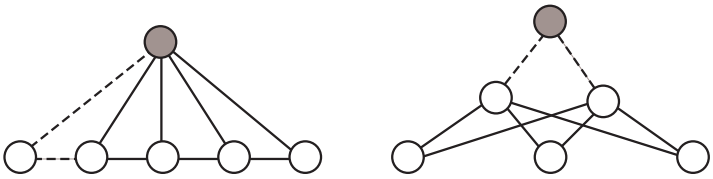


Figure 2.7: Illustration of connectivity not characterizing the robustness of sensor network topologies well. The edge-connectivity of both graphs is 2, and thus, they are equally robust in terms of edge-connectivity. However, when the two dashed edges are removed, only a single vertex is separated from the sink in graph (a), while all of the vertices are separated from the sink in graph (b).

As an example, consider Figure 2.7, which shows two example graphs. Each graph represents a sensor network, in which the operator's objective is to transfer measurement data from the sensor nodes to the sink node, represented by the shaded vertex. Both of these graphs have an edge-connectivity of two, and therefore, they are supposed to be equally robust. However, by removing two edges, we can separate at most one vertex from the sink in graph (a), while we can separate all vertices in graph (b) (the dashed edges represent such attacks in the figure). In other words, a strategic attack removing two edges can only slightly affect graph (a) but can almost completely disable graph (b). Hence, we can hardly say that the networks are equally robust.

Toughness Graph *toughness* is another well-known topology robustness metric with several theoretical results [Bauer et al., 2006]. The toughness of a graph measures the minimum of the ratio between the number of vertices removed and the number of components in the resulting graph. Unfortunately, the problem of computing the toughness of a graph is NP-hard. Hence, it is not very well-suited for general practical use, especially when one is concerned with large graphs.

Strength Graph *strength* is a metric that is very similar to graph toughness. The strength of a graph measures the minimum of the ratio between the number of edges removed and the increase in the number of components in the resulting graph [Cunningham, 1985]. Intuitively, one can think of strength as the "edge-attack version" of toughness. However, unlike toughness, the strength of a graph can be computed in polynomial time. Compared to connectivity, the advantage of graph strength as a robustness metric

is that it considers attacks of various sizes due to the fact that the minimum is taken over all possible edge removal attacks.

2.7.2 Attacker-Model-Driven Studies

While the above metrics have the appeal of simplicity and analytical tractability, it is very hard to argue about how well they capture the notion of robustness. The main reason for this is that they approach the problem of quantifying robustness from a purely graph-theoretic perspective; hence, the corresponding attacker models have to be derived from the metrics and – as a results – usually assume that the strategic nature of the adversary is very limited. In this section, I list examples of related work that approach the problem of studying robustness by starting with an attacker model and "defining" robustness with respect to this model. However, none of these studies consider the simultaneous and strategic decision making of the defender and the attacker, on which the game-theoretic framework of network blocking games is built. Rather, they assume that the attacker will chose from a set of elementary strategies, such as removing the nodes with the highest betweenness, and not anticipate the defender's response.

In [Dall'Asta et al., 2006], the authors study the robustness of networks against various strategies that remove the most central nodes in the network. These strategies are based on ranking nodes using degree, strength, outreach, distance strength, topological betweenness, and weighted betweenness (for the definitions of these metrics, see [Dall'Asta et al., 2006]). To quantify the damage sustained by a network after an attack removing g vertices, three metrics are introduced: the ratio between the total node strength of the damaged network's largest component and that of the intact network, the ratio of total node outreach, and the ratio of total node distance strength. The study shows that complex networks are more fragile than expected form the analysis of topological quantities when the traffic characteristics are taken into account.

In [Estrada, 2006], the authors study the property of graphs being both sparse and highly connected, which is known as "good expansion" (GE). Using spectral graph theory, they introduce a new metric for measuring the good expansion of networks, and classify 51 real-world networks as being GE or non-GE. By comparing the networks based on their robustness against intentional attacks against nodes, the authors argue that being GE and having uniform degree distribution makes networks robust.

In [Holme et al., 2002], the authors study the resilience of complex networks to attacks on vertices and edges. Several existing network models are evaluated with attacking strategies based on removing nodes in the descending order of either the degree or the betweenness centrality. The study shows the Erdös-Rényi random graph model to be the most robust of evaluated models.

2.8 Conclusions

In this chapter, I studied the robustness of network topologies against strategic attacks using a game-theoretic approach, based on a previously proposed framework, called network blocking games. I began with establishing results on the computational complexity of network blocking games in general. First, I showed that the problem of solving a network blocking game is generally NP-hard. While it was obvious from the definition of the game that traditional methods cannot be efficient due to the potentially exponential size of the operator's pure-strategy set, this result shows that even an algorithm that is tailored for network blocking games must have an exponential worst-case running time. Consequently, I turned my attention to a limited subclass of communication models, whose polyhedra have polynomial-size linear characterizations. Even though this requirement might seem like a serious limitation, it turns out that many communication models fall in this class. I showed that, in these communication models, the game can be solved efficiently: the adversary's equilibrium payoff, which we use as our vulnerability metric, can be computed in polynomial time using linear programming tools. Furthermore, if we have an algorithm for computing an operator strategy from an arbitrary element of the polyhedron, which is indeed the case for many models, then we can find an equilibrium strategy profile as well.

Next, I proposed two novel communication models, the All-to-One model and the All-to-All with linear usage model. For both models, I began my analysis with showing that their polyhedra have polynomial-size linear characterizations; hence, the resulting games can be solved efficiently. Furthermore, I also showed that operator strategies can be computed from arbitrary elements of the polyhedra, which allows us to find equilibrium strategy profiles in both models. This means that we can use both

models in practice to quantify the robustness of network topologies and to identify the critical edges that are most likely to be attacked.

The All-to-One model is targeted for studying networks where the nodes have to communicate only with a set of designated nodes, such as sensor or access networks. I first solved a basic model, and then I extended it by introducing non-uniform node values, multiple designated nodes, and a more powerful adversary who can mount attacks against nodes. These extensions allow us to model a wide range of networks and scenarios. Finally, I compared the game-theoretic robustness metric derived from this communication model with a previously proposed graph-theoretic metric, called directed graph strength. I showed that the two metrics are very closely related: if we assume that attacks costs are zero, then the two metrics are actually equivalent. Otherwise, when attack costs are non-zero, the difference is that, in directed graph strength, the adversary maximizes the ratio between her reward and her costs, while in the proposed metric, she maximizes the difference between her reward and her costs.

The All-to-All with linear usage model belongs to a larger family of communication models, which assume that the operator's goal is to ensure that each node is able to communicate with every other node. First, I compared the linear usage function with other, previously proposed usage functions, which were derived from network value functions. Based on comparisons between the values of the usage functions and the critical sets of links identified, I argued that the proposed linear function is at least as realistic as the previous ones. After showing that – opposed to the previous usage functions – the game in the All-to-All model with linear usage can be solved efficiently, I compared the resulting game-theoretic robustness metric with a previously proposed graph-theoretic metric, the Cheeger constant. Similarly to the All-to-One model, I found that the two metrics are closely related: if we assume that attack costs are zero, then the Cheeger constant is equivalent to the proposed metric given that the adversary is limited to attacking only minimal cuts (i.e., partitioning into two connected components). This result can be very surprising considering that the Cheeger constant is NP-hard to compute.

Finally, I extended the general framework of network blocking games by introducing a usage-based cost model and budget limits on the operator. Without this extension, network blocking games assume that the operator takes only security into account and disregards other economic or technical factors. By introducing a budget limit, I remove this restrictive assumption. I proposed two budget-constraint formulations, the Expected Costs and the Maximum Cost Constraints, and studied the complexity of solving the game under each of these constraints. Since I already had that this problem is generally NP-hard, I focused on those models for which the game can be solved efficiently without a budget constraint. First, I showed that the Maximum Costs Constraint unfortunately leads to NP-hard problem in the Supply-Demand, the All-to-One, and the All-to-All with linear usage models. Then, I showed that the Expected Cost Constraint on the other leads to problems that can be solved efficiently.

2.9 Related Publications

- Laszka, A., Szeszlér, D., and Buttyán, L. (2012b). Game-theoretic robustness of many-to-one networks. In *Proceedings of the 3rd International ICST Conference on Game Theory for Networks (GameNets)*, pages 88–98

- Laszka, A., Szeszlér, D., and Buttyán, L. (2012c). Linear loss function for the network blocking game: An efficient model for measuring network robustness and link criticality. In *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)*, pages 152–170

- Laszka, A. and Gueye, A. (2013a). Quantifying All-to-One network topology robustness under budget constraints. In *Proceedings of the joint Workshop on Pricing and Incentives in Networks and Systems (W-PIN+NetEcon)*. ACM

- Laszka, A. and Gueye, A. (2013b). Quantifying network topology robustness under budget constraints: General model and computational complexity. In *Proceedings of the 4th Conference on Decision and Game Theory for Security (GameSec)*, pages 154–174

Chapter 3

Designing Robust WSN Topologies

3.1 Introduction

A usual assumption on wireless sensor networks is that they consist of resource constrained and physically unprotected devices that use wireless channels for communications. These limitations make wireless sensor networks vulnerable to denial-of-service type attacks, such as physical destruction of nodes, exhaustion of their batteries, and jamming of the wireless channels. Such attacks may be addressed at different levels in the system architecture; in this chapter, I focus on mitigating them by controlled node deployment resulting in robust network topologies.

More specifically, I assume that the locations of the sensor nodes are pre-determined by the application requirements, which is indeed the case in most civilian applications of sensor networks (see, e.g., [Welsh, 2010] for some supporting arguments). However, the network operator has some freedom in choosing the location of the sink nodes (or the gateways to some backbone infrastructure). I aim at determining the location of the sinks in such a way that the resulting network be robust against node and link removal attacks (abstracting physical node destruction or exhaustion and jamming, respectively).

In order to be able to compare different sink placements in terms of the robustness of the resulting network, one needs to measure network robustness quantitatively. Instead of the usual approach of using connectivity for this purpose, I use the notion of *persistence*, based on the notion of directed graph strength (which I have already introduced briefly in Section 2.4.3). Roughly speaking, the persistence of a network is defined as the minimum ratio between the cost of an attack and the gain of the attacker, where the cost of an attack is related to the difficulty of removing the attacked links or nodes, and the gain of the attacker is determined by the value of nodes that get disconnected from the sinks as a result of the attack. In Section 2.7, I explained and gave some illustrative examples of why connectivity is not a well-suited metric for quantifying robustness against strategic attacks. In 3.2.2, I will present an additional example showing that persistence can better capture the notion of robustness for wireless sensor networks than connectivity.

Using persistence as the robustness metric, I formalize and study two variants of the sink deployment problem. In the first variant, which I call the sink selection problem, I restrict the set of possible sink locations to the set of sensor node locations (in other words, I allow some of the sensors to be extended with sink functionality). In the second variant, which I call the sink placement problem, I remove this restriction and allow sinks to be placed anywhere in the deployment area. In both variants, I aim at achieving a given level of persistence while minimizing the deployment cost (or – equivalently – maximizing persistence under a given upper bound on the deployment budget). I prove that both sink selection with required persistence and sink placement with required persistence are NP-hard problems. I propose greedy and genetic heuristic algorithms to solve the sink selection problem efficiently, and I show how the problem of sink placement with required persistence can be traced back to the problem of sink selection with required persistence by an efficient search space reduction technique, which may be of independent interest. I also show how any sink selection algorithm, including the proposed heuristic algorithms, can be used to efficiently obtain solutions to the sink placement problem using the proposed search space reduction technique. Finally, I provide experimental results on the performance of the

heuristic algorithms for sink selection and the search space reduction algorithm.

The organization of this chapter is the following. In Section 3.2, I introduce (various versions of) persistence for the purpose of measuring the robustness of networks. In Section 3.3, I formalize the sink selection problem for a given network topology and prove that it is NP-hard. In Section 3.4, I present an integer programming model of the sink selection problem, which can be used to find optimal solutions, and I propose efficient greedy and genetic algorithms as heuristics to compute solutions that are reasonably close to optimal. In Section 3.5, I formalize the sink placement problem and prove that it is NP-hard. In Section 3.6, I propose an algorithm for solving the sink placement problem based on an efficient search space reduction technique. In Section 3.7, I present experimental results on the performance of the heuristic algorithms for sink selection and on the effectiveness of the search space reduction technique. In Section 3.8, I give a brief overview of placement problems in wireless sensor networks and previously proposed solutions. Finally, in Section 3.9, I conclude the chapter.

3.2 Graph Persistence

3.2.1 Definition of Graph Persistence

A common shortcoming – with respect to quantifying the robustness of sensor networks – of all the metrics mentioned in Section 2.7 is that they lack the ability to incorporate the role of sink nodes. The following notion of *persistence* attempts to fill this hiatus. Its definition is obtained by an extension of the notion of *directed graph strength*, which was introduced in [Cunningham, 1985]. However, since it is substantially different from the notion of (undirected) graph strength, I renamed it to avoid ambiguity. I also tailored the definition and the corresponding computation algorithm to the needs of sensor networks: I allow for multiple sinks, attacks against vertices, and undirected edges.

Consider a directed graph G and suppose that a subset of vertices $R \subseteq \mathcal{V}(G)$ is given. Assume that each vertex v needs to communicate with an arbitrary element of R (that is, an element of R should be reachable from v through a directed path in G). Furthermore, there is a weight $s(e)$ assigned to each arc e, which measures the cost of removing (or "attacking") arc e. Finally, there is a weight $d(v)$ assigned to each node v, which measures the loss (or "punishment") if no element of R is reachable from v. When applied to modeling sensor networks, the elements of R correspond to the sink nodes, an edge weight $s(e)$ represents the difficulty of jamming the corresponding link e, and a node weight $d(v)$ represents the importance of information collected by v.

For a subset of arcs $A \subseteq \mathcal{E}(G)$, let $s(A) = \sum_{e \in A} s(e)$, and let $\lambda(A)$ be the sum of the weights $d(v)$ on those vertices v from which no element of R is reachable after deleting all arcs in A. From the attacker's point of view, $s(A)$ and $\lambda(A)$ are the the total cost and total gain of an attack, respectively. Accordingly, the smaller the ratio $\frac{s(A)}{\lambda(A)}$ is, the more efficient an attack removing A is. Therefore, it makes sense to define a robustness metric as the minimum of these ratios.

Definition 8 ((Edge-)persistence)**.** Given a directed graph G, sink nodes $R \subseteq \mathcal{V}(G)$, edge weights $s : \mathcal{E}(G) \to \mathbb{R}^+$, and node weights $d : \mathcal{V}(G) \to \mathbb{R}^+$, the *persistence* (or *edge-persistence*) $\pi(G)$ is defined as

$$\pi(G) = \min \left\{ \frac{s(A)}{\lambda(A)} : A \subseteq \mathcal{E}(G), \lambda(A) > 0 \right\} . \tag{3.1}$$

For an illustrative example, consider again the two graphs from Figure 2.7 (in Section 2.7). Assume that, in both graphs, the shaded vertex is the (single) sink node, all edge weights $s(e)$ and node weights $d(v)$ are 1, and all edges are directed both ways. Then, for both graphs, the minimum in the above definition is attained for the set of edges entering the sink node. Therefore, $\pi(G) = 1$ for graph (a), and $\pi(G) = \frac{2}{5}$ for graph (b). This coincides with the previous observation that graph (a) is intuitively more robust than graph (b), and thus, supports the claim that persistence is a more suitable robustness metric for wireless sensor networks than connectivity.

As mentioned in Section 3.1, attacks against sensor networks are not limited to destroying links between the devices (i.e., removing the edges of the graph), as the devices themselves (i.e., the vertices of the graph) can also be targeted by attacks. Therefore, in order to serve the needs of sensor networks, the above definition needs be extended to allow for the destruction of both edges and vertices.

Definition 9 (Edge-vertex-persistence)**.** Given a directed graph G, sink nodes $R \subseteq \mathcal{V}(G)$, edge and node destruction costs $s : (\mathcal{V}(G) \cup \mathcal{E}(G)) \to \mathbb{R}^+$, and node weights $d : \mathcal{V}(G) \to \mathbb{R}^+$, the *edge-vertex-persistence* $\pi_v(G)$ is defined as

$$\pi_v(G) = \min \left\{ \frac{s(A)}{\lambda(A)} : A \subseteq (\mathcal{V}(G) \cup \mathcal{E}(G)), \lambda(A) > 0 \right\} , \tag{3.2}$$

where $s(A) = \sum_{a \in A} s(a)$, and $\lambda(A)$ is the sum of weights $d(v)$ on those vertices v from which no (remaining) element of R is reachable after deleting all edges and vertices in A.[1]

Fortunately, the problem of computing edge-vertex-persistence can easily be reduced to the problem of computing edge-persistence using *vertex splitting*, a well-known trick in graph theory. First, replace each node v with two nodes v_1 and v_2, add an arc (v_1, v_2) to G. Then, let $s((v_1, v_2)) = s(v)$, let $d(v_1) = d(v)$, let $d(v_2) = 0$, and let $v_2 \in R$ if and only if $v \in R$ was originally true. Finally, replace each original arc (u, v) in G with (u_2, v_1) and set $s((u_2, v_1)) = s((u, v))$. It is fairly easy to see that the edge-persistence of the resulting graph is equal to the edge-vertex-persistence of the original one.

Next, I present another variation of the definition of persistence. In practice, the links of a wireless network can be bidirectional, and an attacker can usually impede both directions with a single attack. Therefore, the definition should be further modified to allow for undirected graphs.

Definition 10 (Undirected persistence)**.** Given an undirected graph G, sink nodes $R \subseteq \mathcal{V}(G)$, edge weights $s : \mathcal{E}(G) \to \mathbb{R}^+$, and node weights $d : \mathcal{V}(G) \to \mathbb{R}^+$, the *undirected persistence* $\pi_u(G)$ is defined as

$$\pi_u(G) = \min \left\{ \frac{s(A)}{\lambda(A)} : A \subseteq \mathcal{E}(G), \lambda(A) > 0 \right\} , \tag{3.3}$$

where $\lambda(A)$ is the sum of the weights $d(v)$ on those vertices v from which there is no undirected path to any element of R after deleting all edges in A.

The problem of computing undirected persistence can be reduced to the problem of computing persistence by replacing each edge with two arcs facing opposite directions, both having the attack cost of the original edge. It is easy to see that the persistence of the resulting graph is equal to the undirected persistence of the original one.

Based on the above arguments, which say that the extended definitions are essentially equivalent to the original one, I restrict my analysis to only consider the edge-persistence of directed graphs for the remainder of this chapter.

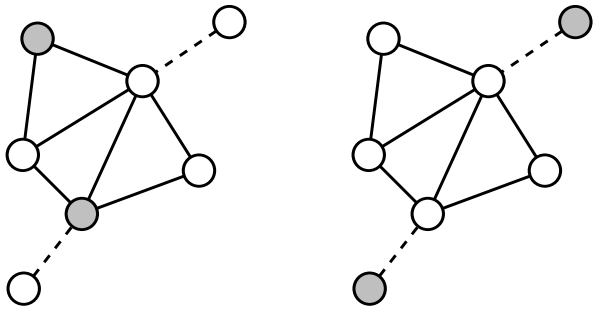
3.2.2 Applications of Persistence

In this subsection, I first present a motivating example, in which selecting sinks based on maximizing persistence instead of connectivity leads to a network that is more robust against attacks. Then, I discuss two other applications of persistence: measuring robustness against random failures and measuring network lifetime.

Robust Sink Selection Example

Figure 3.1 shows two possible sink selections in the same network, one maximizing persistence and one maximizing connectivity. The value of every node and the attack cost of every link is one, and the selected sinks are represented by shaded nodes. The sink nodes, in addition to the links in the figure, are connected to a center as well; therefore, the connectivity of the network measures the number of links that have to be removed in order to disconnect a non-sink node. In selection (a), only one link has to be removed in order to disconnect a node, whereas in selection (b), two links have to be removed. However, the selections have a persistence of 1 and 0.4, respectively. The dashed edges are optimal attacks against the network. In the case of selection (a), only two nodes are separated if the attack is carried out, whereas in the case of selection (b), five nodes are separated. If the network application tolerates the loss of a few nodes, then selection (a) is clearly the better choice.

[1] Naturally, vertices belonging to A also become isolated from R, so they also contribute to the value of $\lambda(A)$.



(a) (b)

Figure 3.1: An example network, where sink selections based on (a) maximizing persistence and (b) maximizing connectivity lead to very different results. Shaded nodes represent the ones that are selected to be sinks. In selection (a), only one link has to be removed in order to disconnect a node, but the persistence of the network is 1, whereas in selection (b), two links have to be removed, but the persistence of the network is only 0.4.

Measuring Robustness against Random Failures

Persistence can be also used to measure the robustness of a network against random link failures under some restrictive assumptions. First, assume that each link e malfunctions with probability $p(e)$ independently of the other links. Next, assign $-\log p(e)$ weight to each edge $e \in \mathcal{E}(G)$ of the graph. Then, the persistence of the graph is

$$\begin{aligned}\pi(G) &= \min\left\{\frac{s(A)}{\lambda(A)} : A \subseteq \mathcal{E}(G),\ \lambda(A) > 0\right\} \\ &= \min\left\{\frac{\sum_{e\in A} -\log p(e)}{\lambda(A)} : A \subseteq \mathcal{E}(G),\ \lambda(A) > 0\right\} \\ &= \min\left\{\frac{-\log \prod_{e\in A} p(e)}{\lambda(A)} : A \subseteq \mathcal{E}(G),\ \lambda(A) > 0\right\} \\ &= \min\left\{\frac{-\log p(A)}{\lambda(A)} : A \subseteq \mathcal{E}(G),\ \lambda(A) > 0\right\}\ , \end{aligned} \tag{3.4}$$

where $p(A)$ is the probability that all links in A malfunction. Intuitively, this means that the probability of an event has to decrease exponentially with its impact.

Measuring Network Lifetime

In [Chang and Tassiulas, 2004], the following model is proposed for measuring the lifetime of wireless sensor networks. Consider a directed graph $G(N, A)$, where N is the set of all nodes and A is the set of all directed links (i, j), where $i, j \in N$. Let S_i be the set of nodes that are in the transmission range of node i. Each node has an initial battery energy of E_i, and the amount of energy consumed when transmitting a packet across link (i, j) is denoted by e_{ij}, where $j \in S_i$.

The goal is to maximize the time T until which the information generated can be delivered to one of the set of gateway nodes $D \subset N$. Let $Q_i(T)$ be the number of packets generated at a sensor node $i \in N \setminus D$ until time T, and let $q_{ij}(T)$ be the total number of packets routed through link $(i, j) \in A$. A time T is feasible if there exists a set of non-negative integers $q_{ij}(T)$ for all links $(i, j) \in A$ satisfying the following two constraints. The first one is the conservation of flow constraint, which can be expressed as

$$\sum_{j:\ i\in S_j} q_{ji}(T) + Q_i(T) = \sum_{k\in S_i} q_{ik}(T),\ \forall i \in N \setminus D\ . \tag{3.5}$$

The second one is the total energy constraint, which can be expressed as

$$\sum_{j\in S_i} e_{ij} q_{ij}(T) \le E_i,\ \forall i \in N \setminus D\ . \tag{3.6}$$

Now, let U be an arbitrary set of nodes. For a node $i \in U$, let ${e_i}^U$ be the minimum energy expenditure for transporting information unit out of U. If there is no outgoing link of i through which information can be forwarded out of U, then ${e_i}^U = \infty$ by convention. The necessary feasibility condition is

$$\sum_{i \in U} Q_i(T) \leq \sum_{i \in U} \frac{E_i}{{e_i}^U} \ . \tag{3.7}$$

Unfortunately, this necessary condition is not sufficient. However, if the energy expenditure through all the outgoing links of a sensor are the same, then the condition is sufficient as well. In this case, maximum lifetime becomes equivalent with undirected-vertex-persistence.

3.2.3 Computing Persistence

It is shown in [Cunningham, 1985] that computing persistence can be performed using a maximum flow algorithm[2]. In particular, assume that besides the input data used above (i.e., G, $R \subseteq \mathcal{V}(G)$, $s : \mathcal{E}(G) \to \mathbb{R}^+$, and $d : \mathcal{V}(G) \to \mathbb{R}^+$) a constant π_0 is also given, which represents a threshold persistence value, and the task is to decide if $\pi(G) \geq \pi_0$ holds.

For any set $X \subseteq \mathcal{V}(G)$, denote by $\delta(X)$ the set of edges leaving X and let $\delta_s(X) = \sum\{s(e) : e \in \delta(X)\}$. It is easy to see that the minimum in the definition of $\pi(G)$ is attained at a set $A = \delta(X)$ for a suitable $X \subseteq \mathcal{V}(G) \setminus R$. Indeed, "spare" edges could be deleted from A without increasing the ratio $s(A)/\lambda(A)$. Of course, $A = \delta(X)$ implies that $s(A) = \delta_s(X)$ and $\lambda(A) = d(X)$, where $d(X) = \sum_{v \in X} d(v))$. Therefore, $\pi(G) \geq \pi_0$ is equivalent to saying that $\delta_s(X) - \pi_0 d(X) \geq 0$ holds for all $X \subseteq \mathcal{V}(G) \setminus R$. By adding $\pi_0 d(\mathcal{V}(G))$ to both sides, we get that $\pi(G) \geq \pi_0$ is equivalent to

$$\delta_s(X) + \pi_0 d(\overline{X}) \geq \pi_0 d(\mathcal{V}(G)) \tag{3.8}$$

for all $X \subseteq \mathcal{V}(G) \setminus R$, where $\overline{X} = \mathcal{V}(G) \setminus X$.

Now, consider the following maximum network flow problem. Add two new nodes, denoted by s^* and t^*, to G. For each $v \in \mathcal{V}(G)$, add a new arc from s^* to v and set its capacity to $\pi_0 d(v)$. For each $v \in R$, add a new arc from v to t^* and set its capacity to infinity. Finally, set the capacity of each original arc of G to $s(e)$. Let us denote the resulting network by G^*. According to the well-known "max-flow-min-cut" theorem of Ford and Fulkerson, the maximum flow from s^* to t^* in the resulting network is equal to the minimum cut capacity, that is, equal to the minimum of the sum of the capacities on arcs leaving a set X, where the minimum is taken over all subsets $X \subseteq \mathcal{V}(G^*)$ for which $s^* \in X$ and $t^* \notin X$. Obviously, the capacity of the cut defined by X is $\delta_s(X) + \pi_0 d(\overline{X})$ if $X \cap R = \emptyset$, and it is infinite otherwise. By comparing this with Equation (3.8) above, we have that $\pi(G) \geq \pi_0$ is equivalent to the existence of a flow of value $\pi_0 d(\mathcal{V}(G))$ from s^* to t^* in G^*. In other words, $\pi(G) \geq \pi_0$ holds if a flow that saturates all arcs leaving s^*.

Consequently, the question of $\pi(G) \geq \pi_0$ can be answered by a maximum flow algorithm. From this, the actual value of $\pi(G)$ (that is, the maximum π_0 for which the above described flow exists) can be determined using binary search, which yields a polynomial time algorithm if all input numerical data is assumed to be integer. In [Cunningham, 1985], a refinement of this approach is also given. It is shown that $\pi(G)$ can be determined by at most $|\mathcal{V}(G)|$ maximum flow computations, even for arbitrary input data.

3.3 The Sink Selection Problem and Its Complexity

Based on the robustness metric $\pi(G)$ defined above, in this section, I formalize the problem of finding the optimal selection of sink nodes in a network with a given topology.

3.3.1 The Sink Selection Problem

Assume that assigning the sink role to a node v has some cost $c(v)$ resulting from, for example, establishing an external connection with the node, regularly visiting the node for data collection, etc. Let us

[2]In this subsection I build on the basics of network flow theory; the required background can be found in most introductory graph theory textbooks.

call this cost the *selection cost* of the sink, and assume that sink selection costs are additive (i.e., the cost of assigning the sink role to a set of nodes is simply the sum of the selection costs of the nodes in the set). Furthermore, assume that the network topology is given and our task is only to select the sink vertices such that the persistence of the resulting network configuration is above a given threshold, while the total selection cost of the sink nodes is minimized. This models the design of a wireless sensor network with strict security requirements, but with a flexible budget.

Based on the above definitions, the sink selection problem is formulated as follows.

Definition 11 (Sink Selection with Required Persistence)**.**

INSTANCE: Directed graph G, edge weights $s : \mathcal{E}(G) \to \mathbb{R}^+$, node weights $d : \mathcal{V}(G) \to \mathbb{R}^+$, sink selection costs $c : \mathcal{V}(G) \to \mathbb{R}^+$, and required persistence $\pi_0 \in \mathbb{R}^+$.

SOLUTION: A subset $R \subseteq \mathcal{V}(G)$ such that the persistence $\pi(G)$ of G is at least π_0 with R as its sink nodes.

MINIMIZE: Selection cost of subset R, i.e., $\sum_{v \in R} c(v)$.

Obviously, the variant of the sink selection problem where a limit on the total sink selection cost is given and the persistence of the configuration is to be maximized is also sensible. I disregard this variant of the problem, and restrict my analysis to mentioning that any algorithm for solving one of the two variants can also be used to solve the other one using binary search.

3.3.2 Complexity of the Sink Selection Problem

Now, I study the computational complexity of the sink selection problem. The motivation for this question is that, in practice, the number of nodes in WSNs ranges from dozens [Skalka and Frolik, 2013] to hundreds [Kerkez et al., 2012, Baggio, 2005]; hence, sink selection algorithms need to scale well. As the main result of this section, I prove that the Sink Selection problem is NP-hard. To this end, I show that the Minimum Set Cover Problem, one of the well-known NP-hard problems, can be reduced to it. The (decision version of the) Minimum Set Cover problem is defined as follows.

Definition 12 (Minimum Set Cover)**.**

INSTANCE: A finite set $U = \{u_1, u_2, \ldots, u_n\}$, a collection of its subsets $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ ($A_i \subseteq U$ for all $1 \leq i \leq m$), and a positive integer r.

TASK: Decide if it is possible to choose at most r subsets from $\mathcal{A}$ that cover U; that is, if the subsets $A_{i_1}, A_{i_2}, \ldots, A_{i_p}$ can be chosen such that $p \leq r$ and $\bigcup_{j=1}^{p} A_{i_j} = U$. (The chosen subsets A_{i_j} are said to form a *set cover of size p.*)

The decision version of the Sink Selection problem is defined in the most natural way as follows.

Definition 13 (Sink Selection with Required Persistence (decision version))**.**

INSTANCE: Directed graph G, edge weights $s : \mathcal{E}(G) \to \mathbb{R}^+$, node weights $d : \mathcal{V}(G) \to \mathbb{R}^+$, sink selection costs $c : \mathcal{V}(G) \to \mathbb{R}^+$, required persistence $\pi_0 \in \mathbb{R}^+$ and maximum cost $c_0 \in \mathbb{R}^+$.

TASK: Decide if a subset $R \subseteq \mathcal{V}(G)$ exists such that $\pi(G, R) \geq \pi_0$ and $\sum_{v \in R} c(v) \leq c_0$.

The following theorem shows that the Sink Selection with Required Persistence problem is NP-hard by reducing it to the Minimum Set Cover problem.

Theorem 12. *The Sink Selection with Required Persistence problem is NP-complete.*

Proof. The problem is obviously in NP since, if R is given, then checking if $\pi(G, R) \geq \pi_0$ can be done in polynomial time according to [Cunningham, 1985] (and checking if $\sum_{v \in R} c(v) \leq c_0$ is trivial).

As mentioned above, I show NP-hardness by reducing the Minimum Set Cover problem to the Sink Selection problem. So, assume that an instance of the Minimum Set Cover problem (that is, $U = \{u_1, \ldots, u_n\}$, $\mathcal{A} = \{A_1, \ldots, A_m\}$ and r) is given. Obviously, we can assume that $\bigcup_{j=1}^{m} A_j = U$ (otherwise there exists no set cover at all) and that $r \leq m$ (otherwise the problem is trivial). From this, I construct an instance of the Sink Selection problem in the following way.

1. Let $\mathcal{V}(G) = U \cup \mathcal{A}$; that is, to each element $u_i \in U$ and to each given subset $A_i \in \mathcal{A}$ corresponds a vertex of G.

2. Let $\mathcal{E}(G) = \{(u_i, A_j) : u_i \in U, A_j \in \mathcal{A}, u_i \in A_j\}$; that is, whenever $u_i \in A_j$ holds for an element $u_i \in U$ and $A_j \in \mathcal{A}$, introduce a directed edge from u_i to A_j in G. (Consequently, G is a directed bipartite graph.)

3. Let $d(u_i) = 1$ and $c(u_i) = r + 1$ for every $u_i \in U$, let $d(A_j) = 0$ and $c(A_j) = 1$ for every $A_j \in \mathcal{A}$ and let $s(e) = 1$ for every $e \in \mathcal{E}(G)$.

4. Finally, let $\pi_0 = 1$ and $c_0 = r$.

We have to show that there exists a set cover $\mathcal{A}_0 \subseteq \mathcal{A}$ of size at most r if and only if there exists a sink selection $R \subseteq \mathcal{V}(G)$ such that $\pi(G, R) \geq \pi_0 = 1$ and $\sum_{v \in R} c(v) \leq c_0 = r$.

First assume that a sink selection R with the above properties is given. Since the costs of the vertices in U are all $r + 1$, selection R contains vertices only from $\mathcal{A}$. I claim that $\mathcal{A}_0 = R$ forms a set cover. For the sake of contradiction, suppose that this is not true. Then, there exists an element $u_i \in U$ not covered by R; that is, $u_i \notin \bigcup\{A_j : A_j \in R\}$. Consequently, no element of R is reachable from u_i in G; therefore, $\lambda(\emptyset) \geq 1$ holds by $d(u_i) = 1$. Since $s(\emptyset) = 0$ holds obviously, this implies that $\pi(G, R) = 0$ (by $\frac{s(\emptyset)}{\lambda(\emptyset)} = 0$), which is a contradiction. So R has to be a set cover, and its size is obviously at most r (since $c(A_j) = 1$ for every $A_j \in A$).

Now, assume that a set cover $\mathcal{A}_0$ of size $p \leq r$ is given, and let $R = \mathcal{A}_0$. I claim that R is a sink selection in G that satisfies the above requirements. Obviously, $\sum_{v \in R} c(v) = p \leq r$. To show $\pi(G, R) \geq 1$, let $Y \subseteq \mathcal{E}(G)$ be any subset of edges. Then, we need to prove that $s(Y) \geq \lambda(Y)$. It is easy to see that $s(Y) = |Y|$ and $\lambda(Y) = |X|$, where $X \subseteq U$ is the set of vertices in U from which no element of R is reachable after removing Y.[3] So, $|Y| \geq |X|$ is to be proved. However, this immediately follows from the fact that since R is a set cover, there exists an $A_j \in R$ for every $u_i \in X$ such that $u_i \in A_j$, so we can assign a separate edge in Y to each element of X (namely, the one that goes from u_i to A_j). □

I remark that the construction of the above proof could be modified in the following way: instead of setting $c(u_i) = r + 1$ for every $u_i \in U$, we could replace u_i by $r + 1$ vertices, each with a cost of 1 (and each connected to A_j, whenever $u_i \in A_j$). It is easy to verify that the proof goes through with the modified construction as well, which shows that the Sink Selection problem remains to be NP-complete even under the restriction that $c(v) = 1$ for every $v \in \mathcal{V}(G)$.

The essence of the above proof is that, in the described construction, set covers of size r and proper sink selections of total cost r are basically the same. This simple fact has an important consequence on the approximability of the Sink Selection problem.

Theorem 13. *Assuming that $P \neq NP$, there exists a constant $c > 0$ such that there is no polynomial-time algorithm that finds a sink selection of total cost at most $c \log\log|\mathcal{V}| \cdot OPT$, where OPT denotes the minimum total cost of a sink selection.*

Recall that $P \neq NP$ is a widely accepted conjecture; if this were not true, then there would exist a polynomial-time algorithm for every NP-hard problem. An obvious corollary of the above theorem is that there is no constant-factor approximation algorithm for the sink selection problem unless $P = NP$.

Proof. Assume that there is an algorithm P that finds a sink selection of total cost at most $c \log\log|\mathcal{V}| \cdot OPT$ for some constant c. Then, if an instance $U = \{u_1, \ldots, u_n\}$, $\mathcal{A} = \{A_1, \ldots, A_m\}$ of the (optimization version of the) Minimum Set Cover problem is given, P can be applied on the construction given in the above proof. Then, if v denotes the number of vertices in the constructed graph, $v = n + m \leq n + 2^n \leq 2^{n+1}$ holds. Since, as remarked above, sink selections in the constructed graph are essentially the same as possible solutions of the given Minimum Set Cover instance, OPT also denotes the minimum size of a set cover for this. Therefore, P finds a set cover of size at most $c \log\log 2^{n+1} OPT = c \log(n+1) OPT$. However, in [Raz and Safra, 1997], it was proved that, if $P \neq NP$, then there is no polynomial-time algorithm that finds a set cover of size at most $c_1 \log n \cdot OPT$, where $c_1 > 0$ is an appropriate constant. This immediately implies the existence of the required c. □

[3] Observe that X is not the set of *all* vertices of G from which R is not reachable after removing Y: vertices in $\mathcal{A} \setminus R$ also have this property. However, these do not contribute to $\lambda(Y)$ as they have a weight of 0.

3.4 Algorithms for Solving the Sink Selection Problem

In this section, I show how an optimal sink selection can be found by solving an integer program, and I also introduce more efficient heuristic algorithms, which approximate the optimal solution reasonably well. Performance evaluation and quantitative comparison of these algorithms is provided in Section 3.7.1.

3.4.1 Integer Programming Model for the Sink Selection Problem

To formulate the sink selection problem as an integer program, I assign a binary variable $r(v)$ to each node v: the value of $r(v)$ is 1 if v belongs to R, and 0 otherwise. The formulation relies on the construction presented in Section 3.2.3: $\pi(G) \geq \pi_0$ is true if and only if there exists a flow in the network G^* described there that saturates all edges (s^*, v). Correspondingly, I assign a variable $f(e)$ to each edge $e \in \mathcal{E}(G^*)$ to measure the flow on e. As it is natural in network flow theory, all the constraints ensuring that f is a flow (that is, capacity constraints and flow preservation constraints) can be formulated as linear constraints in a straightforward way.

The only difference from the construction described in Section 3.2.3 is that the set of sink nodes R is not known in advance. Therefore, I assume that an arc from v to t^* exists from each node $v \in \mathcal{V}(G)$, and ensure that the capacity of the arc (v, t^*) is ∞ for sink nodes and 0 for non-sink nodes. I achieve this by imposing an inequality $f((v, t^*)) \leq \mathit{bignum} \cdot r(v)$ on each edge (v, t^*), where bignum is a sufficiently large constant (for example, we can let $\mathit{bignum} = d(\mathcal{V}(G))$, since the sum capacity of all arcs leaving s^* is greater than or equal to the maximum flow value even if all vertices are assumed to be sinks).

With respect to the above, the integer program is the following.

$$\text{Minimize} \sum_{v \in \mathcal{V}(G)} c(v) \cdot r(v) \tag{3.9}$$

subject to

$$\forall v \in \mathcal{V}(G): \quad f((v, t^*)) \leq \mathit{bignum} \cdot r(v) \tag{3.10}$$

$$\forall e \in \mathcal{E}(G): \quad f(e) \geq 0 \tag{3.11}$$

$$\forall e \in \mathcal{E}(G): \quad f(e) \leq s(e) \tag{3.12}$$

$$\forall v \in \mathcal{V}(G): \quad \sum_{(u,v) \in \mathcal{E}(G^*)} f((u, v)) = \sum_{(v,u) \in \mathcal{E}(G^*)} f((v, u)) \tag{3.13}$$

$$\forall v \in \mathcal{V}(G): \quad f((s^*, v)) \geq \pi_0 \cdot d(v)\ , \tag{3.14}$$

where $r(v) \in \{0, 1\}$ for all $v \in \mathcal{V}(G)$, $f(e) \in \mathbb{R}$ for all $e \in \mathcal{E}(G^*)$, bignum is a sufficiently large number, $s((u, v))$ is the weight of edge (u, v), $d(v)$ is the weight of node v, $c(v)$ is the selection cost of node v, and π_0 is the required persistence.

Constraints (3.11), (3.12), and (3.13) ensure that f is a flow: Constraints (3.12) and (3.13) correspond to capacity and flow preservation constraints, respectively. Note that capacity constraints are not imposed on (s^*, v) type arcs (as opposed to what was said in Section 3.2.3); these can be omitted, as they obviously cannot affect the optimum solution. On the other hand, Constraint (3.14) ensures that all edges (s^*, v) are saturated. Finally, the role of Constraint (3.10) was already explained above.

Unfortunately, the above integer program does not give us an efficient algorithm for solving the sink selection problem. However, it does enable us to obtain the optimum solution for relatively small problem instances, and consequently, it can be used to test the heuristics presented in the following subsections. Later, I will also use it to compare the proposed search-space reduction technique with other techniques.

3.4.2 Greedy Algorithm

The exponential time complexity of solving the above integer program limits its practical applicability. For this reason, in this subsection, I propose an efficient greedy algorithm as a heuristic approach for finding sub-optimal but reasonably good solutions to the sink selection problem.

The greedy algorithm starts with the set of selected sinks R as the empty set. In each step, a new vertex v is added to R. The next vertex v is chosen in a simple but sensible way: v is chosen such that the ratio between the gain in persistence by adding v to R and the selection cost $c(v)$ is maximum. The

algorithm stops when the persistence $\pi(G)$ of the network with the set of sinks R reaches the required persistence π_0.

To formally describe the algorithm, denote by $\pi(G, R)$ the persistence of the network G with R as its set of sink nodes. Then, the greedy algorithm for sink selection with required persistence is the following.

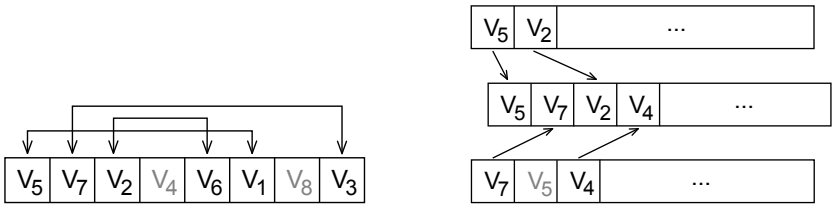
1. Let $R := \emptyset$.
2. Let $v \in \mathcal{V}(G) \setminus R$ be a vertex for which the maximum
$$\max_{v \in \mathcal{V}(G) \setminus R} \frac{\pi(G, R \cup \{v\}) - \pi(G, R)}{c(v)}$$
is attained, and let $R := R \cup \{v\}$.
3. If $\pi(G, R) \geq \pi_0$, then return R; otherwise, continue from Step 2.

I emphasize that – obviously – the above algorithm runs in polynomial time, since it makes at most $|\mathcal{V}(G)|$ iterations and each iteration requires at most $|\mathcal{V}(G)|$ computations of persistence.

3.4.3 Genetic Algorithm

For a higher number of nodes, even the greedy algorithm's computational complexity may be too high. Therefore, in this subsection, I propose a genetic algorithm as a more efficient alternative to the greedy algorithm.

In the proposed genetic algorithm, an individual member of the population represents a solution to the sink selection problem. To make the evolutionary process more efficient, I have chosen to encode not only the set of nodes which are selected, but also a preference for each node, including those that are not selected by the given solution. These preferences can be represented by the order in which the nodes are picked until the the given persistence value is reached. If this representation is used, then it is not necessary to record which nodes are selected, as the number of nodes that need to be selected can be determined from the order of preference. Therefore, the solution domain is simply the set of node sequences.



(a) Mutation (b) Crossbreeding

Figure 3.2: Illustration of how (a) mutation and (b) crossbreeding are implemented.

I have chosen the fitness of each solution to be – quite naturally – the selection cost of the given solution. I have implemented two genetic operators (see Figure 3.2 for illustrations): mutation and crossover. A mutation is a reordering of the nodes, which is achieved by selecting a fixed number of random pairs from the sequence, and swapping the elements of each pair. A crossover is the combination of two node sequences, which is achieved by iteratively picking the most preferred, but not yet picked node from the two sequences in turns.

As the genetic operators are very simple, the performance of the algorithm depends primarily on the efficient evaluation of the fitness function. The following algorithm is proposed for this.

1. Add a source vertex s^* and a sink vertex t^* to the graph. For each $v \in \mathcal{V}(G)$, add a new arc from s^* to v and set its capacity to $\pi_0 \cdot d_v$. Let $n = 1$.
2. Add a new arc from the nth vertex to t^* with infinity capacity.
3. Find a maximum flow in this network by augmenting the flow values of the previous iteration.

4. If the maximum flow is at least $\sum_{v \in \mathcal{V}(G)} \pi_0 \cdot d_v$, then the cost of the given solution is $\sum_{i=1}^{n} c_i$. Otherwise, let $n := n + 1$ and continue from Step 2.

The construction of the graph in Step 1 is the same as in Section 3.2.3, except for the omission of the arcs from sink nodes to the super sink, as the set of sink nodes is not known in advance in this case. In Step 3, the maximum flow of the previous iteration is reused, which makes the algorithm very efficient. The complexity of determining the number of necessary nodes is equal to the complexity of testing whether a graph has a given persistence, which is very simple compared to computing persistence. Therefore, we can efficiently create and evaluate a large number of solutions. The genetic algorithm terminates after a fixed number of generations.

3.5 The Sink Placement Problem and Its Complexity

In this section, I relax some of the previous restrictions on the design of the deployment configuration, and allow sinks to be placed anywhere. However, I still consider the placement of non-sink nodes and the links between them to be given. Therefore, the goal is to design a robust sink placement for a given network topology. By sink placement, I mean a set of locations where sink nodes have to be placed.

3.5.1 The Sink Placement Problem

Before formulating the sink placement problem, I first have to establish a sink node model and define the persistence of a placement.

Definition 14 (Persistence of a Placement)**.** Let G be a directed geometric graph, where $\mathcal{V}(G)$ is a set of points in the Euclidean plane and $\mathcal{E}(G)$ is an arbitrary subset of $V^2(G)$ (i.e., the edges of the graph do not have to follow any geometric rule). Let $s: \mathcal{E}(G) \to \mathbb{R}^+$ be edge weights, let $d: \mathcal{V}(G) \to \mathbb{R}^+$ be node weights, and let $D \in \mathbb{R}^+$ be a fixed sink transmission radius. The *persistence of a placement* R, where R is a set of points in the Euclidean plane, for graph G, denoted by $\pi_p(G, R)$, is the persistence of the graph G' with R as its sinks, where

1. $\mathcal{V}(G') = \mathcal{V}(G) \cup R$,
2. $\mathcal{E}(G') = \mathcal{E}(G) \cup \{(v, r) \ : \ v \in \mathcal{V}(G) \wedge r \in R \wedge \mathit{distance}(v, r) \leq D\}$,
3. $\forall_{(v,r) \ \in \ \mathcal{V}(G) \times R}: \ s(v, r) = 1$,
4. $\forall_{r \in R}: \ d(r) = 0$.

This definition establishes a model for sink nodes, in which sinks have uniform transmission radii (2) and zero value (4), and the links connected to them have uniform weights (3). This is a realistic model for sensor networks, where typically a large number of similar nodes are deployed. The direction of the links connected to the sinks can be surprising at first, since the goal is to model wireless networks, where links are bidirectional. However, we have seen that the undirected persistence of a graph is equal to its directed persistence (with each undirected edge being replaced by two directed edges facing opposite directions). In addition, edges leaving sinks can be omitted, as no traffic has to be carried from a sink. Therefore, the direction of the sink edges is appropriate for the model. Formulating the persistence of a placement this way simplifies the algorithm presented in Section 3.6.2.

Using the above definition of the persistence of a placement, the robust sink placement problem can be formulated as follows.

Definition 15 (Sink Placement with Required Persistence)**.**

INSTANCE: Directed graph G, where $\mathcal{V}(G)$ is a set of points in the Euclidean plane, edge weights $s: \mathcal{E}(G) \to \mathbb{R}^+$, node weights $d: \mathcal{V}(G) \to \mathbb{R}^+$, sink transmission radius D, and required persistence $\pi_0 \in \mathbb{R}^+$.

SOLUTION: A set of points R in the Euclidean plane such that $\pi_p(G, R) \geq \pi_0$.

MINIMIZE: The number of sinks required by the placement, i.e., $|R|$.

Note that, in the above definition, the set of feasible solutions is defined using $\pi_p(G, R)$, which denotes the persistence of a placement R.

3.5.2 Complexity of the Sink Placement Problem

As the sink placement problem is a "generalization" of the sink selection problem with uniform selection costs, it is we can expect it to be NP-hard.

To prove that the placement problem is indeed NP-hard, I introduce the decision version of the Sink Placement with Required Persistence problem.

Definition 16 (Sink Placement with Required Persistence (decision version))**.**

INSTANCE: Directed graph G, where $\mathcal{V}(G)$ is a set of points in the Euclidean plane, edge weights $s : \mathcal{E}(G) \to \mathbb{R}^+$, node weights $d : \mathcal{V}(G) \to \mathbb{R}^+$, sink transmission radius D, required persistence $\pi_0 \in \mathbb{R}^+$ and maximum number of sinks $c_0 \in \mathbb{R}^+$.

TASK: Decide if it is possible to place at most c_0 sinks in the Euclidean plane such that $\pi_p(G, R) \geq \pi_0$.

Theorem 14. *The Sink Placement with Required Persistence problem is NP-hard.*

Proof. I show NP-hardness by reducing the problem of Sink Selection with Required Persistence and uniform selection costs to the problem of Sink Placement with Required Persistence. So, assume that an instance of the Sink Selection problem (that is, G, $s : \mathcal{E}(G) \to \mathbb{R}^+$, $d : \mathcal{V}(G) \to \mathbb{R}^+$, c_0 and π_0) is given. We can assume that $\pi_0 < \frac{s(\mathcal{E}(G))}{\min_{v \in \mathcal{V}(G): d(v) > 0}\{d(v)\}}$. Otherwise, the only feasible solution to the problem is to select every $v \in \mathcal{V}(G) : d(v) > 0$, and checking whether the cost of this selection is lower than c_0 can be done in polynomial time. From the instance of the Sink Selection problem, we construct an instance of the Sink Placement problem (that is, G', $s' : \mathcal{E}(G') \to \mathbb{R}^+$, $d' : \mathcal{V}(G') \to \mathbb{R}^+$, D, c_0' and π_0') in the following way.

1. Let $\mathcal{V}(G') = \mathcal{V}(G)$, $\mathcal{E}(G') = \mathcal{E}(G)$, $d' = d$ and $c_0' = c_0$.
2. Let $D = 1$ and position the vertices of G' on the plane such that the distance between each pair of vertices is greater than 2.
3. Let $s' = \beta s$, where $\beta = \frac{\min_{v \in \mathcal{V}(G): d(v) > 0}\{d(v)\}}{s(\mathcal{E}(G)) \sum_{v \in \mathcal{V}(G)} d(v)}$.
4. Finally, let $\pi_0' = \beta \pi_0$.

Let $\lambda_G(A)$ and $\lambda_{G'}(A)$ denote the value of $\lambda(A)$ in G and G', respectively. I have to show that there exists a selection $R \subseteq \mathcal{V}(G)$ such that $\pi(G, R) \geq \pi_0$ and $\sum_{r \in R} c(r) \leq c_0$ if and only if there exists a sink placement R' such that $\pi_p(G', R') \geq \pi_0'$ and $|R'| \leq c_0'$.

First, assume that a sink placement R' with the above properties is given. Let R be the subset of vertices that have at least one sink in their proximity. I claim that R is a feasible selection, i.e., $\pi(G, R) \geq \pi_0$. For the sake of contradiction, suppose that this is not true. Then, there exists an $A \subseteq \mathcal{E}(G)$ attack for which $\frac{s(A)}{\lambda_G(A)} < \pi_0$ holds. If a vertex cannot reach any $r \in R$ sink in G when A is removed, then the same vertex cannot reach any $r' \in R'$ sink in G' either, as any path leading to a sink necessarily goes through an $r \in R$. Therefore, we have $\lambda_G(A) \leq \lambda_{G'}(A)$. Since $s'(A) = \beta s(A)$, this implies the contradiction $\frac{s'(A)}{\lambda_{G'}(A)} \leq \frac{s'(A)}{\lambda_G(A)} = \frac{\beta s(A)}{\lambda_G(A)} < \beta \pi_0 = \pi_0'$.

Now, I have to show that $\sum_{r \in R} c(r) \leq c_0$. Since the distance between each pair of vertices is greater than 2, at most one vertex is connected to every sink. Therefore, R has at most $|R'|$ vertices and $\sum_{r \in R} c(r) = \sum_{r \in R} 1 \leq |R| \leq c_0$ holds.

Second, assume that a sink selection in R with the above properties is given. Let R' be a sink placement constructed from the positions of the vertices in R. I claim that R' is a feasible placement, i.e., $\pi_p(G', R') \geq \pi_0'$. For the sake of contradiction, suppose that this is not true. Then, there is an $A \subseteq \mathcal{E}(G')$ attack for which $\frac{s'(A)}{\lambda_{G'}(A)} < \pi_0'$ holds. First, observe that A does not contain any edges connected to sinks, i.e., $A \subseteq \mathcal{E}(G)$. Otherwise, the $s'(A)$ cost of the attack would be at least 1 and, since $\lambda_{G'}(A) \leq \sum_{v \in \mathcal{V}(G)} d(v)$, the ratio $\frac{s'(A)}{\lambda_{G'}(A)} \geq \frac{1}{\sum_{v \in \mathcal{V}(G)} d(v)} = \beta \frac{s(\mathcal{E}(G))}{\min_{v \in \mathcal{V}(G): d(v) > 0}\{d(v)\}} > \beta \pi_0 = \pi_0'$. If a vertex v cannot reach any $r' \in R'$ sink in G' when A is removed, then v cannot reach any $r \in R$ sink in G either when A is removed. Otherwise, there would be a path in G from v to an $r \in R$, but then this path could be supplemented with an edge leading to a sink $r' \in R'$ and, therefore, v could reach an $r' \in R'$ sink even when A is removed. Consequently, $\lambda_{G'}(A) \leq \lambda G(A)$ has to hold. Since $s(A) = \frac{1}{\beta} s'(A)$, this implies the contradiction $\frac{s(A)}{\lambda_G(A)} \leq \frac{s(A)}{\lambda_{G'}(A)} = \frac{1}{\beta} \frac{s'(A)}{\lambda_{G'}(A)} < \frac{1}{\beta} \pi_0' = \pi_0$.

Finally, I have to show that $|R'| \leq c_0$. This is obvious, as $|R'| = |R| = \sum_{r \in R} 1 = \sum_{r \in R} c(r) \leq c_0$. □

3.6 Algorithm for Solving the Sink Placement Problem

In this section, I introduce a technique for reducing the infinite search space of possible placements to a finite set, which always includes an optimal solution. I also present an algorithm that can be used to find an optimal placement in the reduced search space using an arbitrary algorithm for finding an optimal selection.

3.6.1 Search Space Reduction Technique

To simplify the subsequent definitions, I first introduce the concept of single sink coverable sets.

Definition 17 (Single Sink Coverable Set)**.** A set of points W in the Euclidean plane is *single sink coverable* for a transmission radius D, if there exists a point r in the plane such that $distance(w, r) \leq D$ for every $w \in W$ (i.e., the points can be covered by a disk of radius D).

I reduce the infinite search space of possible placements by restricting the positions of the sink nodes to a set of candidate locations. Contrary to most candidate location sets proposed in the literature, the subsets of the proposed set always include an optimal solution. For this reason, I call it an optimal set of candidate locations.

Definition 18 (Optimal Set of Candidate Locations)**.** Given a geometric graph G and a sink transmission radius D, an *optimal set of candidate locations* $R_{\text{candidate}}$ is a set of positions which includes exactly one position covering every inclusion-maximal single sink coverable subset of $\mathcal{V}(G)$ for D (i.e., includes a location for every single sink coverable set of node positions that is not a subset of a larger single sink coverable set).

I will prove that (1) the subsets of an optimal set of candidate locations always include an optimal placement and that (2) an optimal set of candidate locations can be found in polynomial time.

Theorem 15. *The subsets of an optimal set of candidate locations include an optimal placement for every persistence requirement.*

Proof. Suppose that R' is an optimal sink placement for a given persistence requirement. Let the set of nodes which are covered by an $r' \in R'$ node be denoted by $N(r')$. Obviously, $N(r')$ is a single sink coverable set. If $N(r')$ is an inclusion-maximal single sink coverable set, then there is an $r \in R_{\text{candidate}}$ which covers exactly $N(r')$; otherwise, there is an $r \in R_{\text{candidate}}$ which covers all nodes covered by r' and some others as well. It is easy to see that if we substitute r' with r, then the solution still satisfies the persistence requirement.

Therefore, by substituting every $r' \notin R_{\text{candidate}}$ with an appropriate $r \in R_{\text{candidate}}$, we can obtain an $R \subseteq R_{\text{candidate}}$ which satisfies the persistence requirement and for which $|R| = |R'|$. □

Before introducing a polynomial time algorithm for finding an optimal set of candidate locations, I first prove that the size of the optimal set is polynomial in the number of nodes.

Theorem 16. *There exists an optimal set of candidate locations with cardinality less than or equal to* $|\mathcal{V}(G)|^3$.

Proof. First, I prove that the number of candidate locations is less than or equal to the number of all smallest disks enclosing subsets of $\mathcal{V}(G)$. If a set of points is single sink coverable, then the radius of the smallest disk enclosing the set of points is at most D. Now, assign to each inclusion-maximal single sink coverable set the center of the smallest disk enclosing the set. Clearly, no disk center (i.e., point in the plane) is assigned to multiple sets; otherwise, these sets could all be covered by a disk of radius D around that center and they would not be inclusion-maximal. Therefore, the number of inclusion-maximal single sink coverable sets is less than or equal to the number of all smallest disks enclosing subsets of $\mathcal{V}(G)$.

Second, I prove that the latter is at most $|\mathcal{V}(G)|^3$. It is well-known (see [Welzl, 1991]) that the smallest disk enclosing a set of points is unique and

- the endpoints of a diameter are members of the set
- or three points of the set forming an acute triangle are on the circumference.

Therefore, by enumerating

- the midpoints of all line segments formed by point pairs in $\mathcal{V}(G)$
- and the centers of all circumcircles of acute triangles formed by points of $\mathcal{V}(G)$,

we can enumerate all possible centers of smallest disks enclosing subsets of $\mathcal{V}(G)$. The number of pairs is $\binom{|\mathcal{V}(G)|}{2}$ and the number of triangles is at most $\binom{|\mathcal{V}(G)|}{3}$. Therefore, the number of candidate locations is at most $|\mathcal{V}(G)|^3$. □

The proposed algorithm for finding an optimal set of candidate locations is based on the ideas behind the above proof. Given a set of node positions $\mathcal{V}(G)$ and a sink transmission radius D, the following algorithm finds an optimal set of candidate locations.

1. Let $R_{\text{circumcenters}} = \emptyset$.
2. For every $\{p_1, p_2\} \in \mathcal{V}(G)^2$, add the midpoint of the line segment p_1, p_2 to $R_{\text{circumcenters}}$.
3. For every $\{p_1, p_2, p_3\} \in \mathcal{V}(G)^3$, add the center of the circumcircle of p_1, p_2, p_3 to $R_{\text{circumcenters}}$ if p_1, p_2, p_3 form an acute triangle.
4. Let $R_{\text{candidate}} = \emptyset$.
5. For every $r \in R_{\text{circumcenters}}$,
   (a) if $\exists r' \in R_{\text{candidate}} : N(r) \subseteq N(r')$ then continue with the next iteration,
   (b) otherwise, for every $r'' \in R_{\text{candidate}}$, if $N(r'') \subseteq N(r)$ then remove r'' from $R_{\text{candidate}}$
   (c) and add r to $R_{\text{candidate}}$.

Obviously, the above algorithm runs in polynomial time. The correctness of the algorithm follows readily from the proof of Theorem 16.

In practice, the maximal density of the nodes (i.e., the maximal number of nodes in a given area) is usually limited. It is easy to see that the number of candidate locations is $O(|\mathcal{V}(G)|)$ in this case. For every node, the number of nodes nearer than $2 \cdot D$ is limited; therefore, the number of maximal single sink coverable sets containing the node is less than a certain constant. Since this holds for each $|\mathcal{V}(G)|$ node, the above claim is obviously true. Experimental results in Subsection 3.7.2 show that, in practice, the number of candidate locations is indeed linear in the number of nodes.

3.6.2 Placement Algorithm

Based on the proposed search space reduction technique, in this subsection, I introduce an algorithm which solves the problem of sink placement with required persistence using an existing algorithm for solving the problem of sink selection with required persistence.

Let us assume that each candidate location is used only once, i.e., no two sink nodes are to be placed at the same location. This is a realistic assumption as more than one sink node is required at a single location only if the sink nodes themselves are too vulnerable. This would indicate that the required persistence goal is not met because the devices are not robust enough, not because the network topology is vulnerable.

Given an algorithm $\mathcal{A}$ for sink selection, the following algorithm solves the sink placement problem.

1. Find an optimal set of candidate locations $R_{\text{candidate}}$.
2. Let G' be a graph defined as the following:
   - $\mathcal{V}(G') := \mathcal{V}(G) \cup R_{\text{candidate}}$
   - $\mathcal{E}(G') := \mathcal{E}(G) \cup \{(v, r) \ : \ v \in \mathcal{V}(G) \wedge r \in R \wedge \mathit{distance}(v, r) \leq D\}$
   - $\forall_{(v,r) \ \in \ \mathcal{V}(G) \times R} \ s(v, r) := 1.$
   - $\forall_{v \in \mathcal{V}(G)} \ c(v) := \infty$
     $\forall_{v \in R_{\text{candidate}}} \ c(v) := 1$ and $d(v) = 0$

3. Find an optimal sink selection R_{opt} in G' with required persistence π_0 using $\mathcal{A}$.

4. Output R_{opt} as the optimal set of points for sink placement.

First, I prove that the set of feasible selections in G' is equal to the set of feasible placements restricted to $R_{\text{candidate}}$ in G.

Theorem 17. *Given an $R \subseteq R_{\text{candidate}}$, $\pi(G', R) \geq \pi_0$ if and only if $\pi_p(G, R) \geq \pi_0$.*

Proof. The graph G' constructed in the above algorithm and the graph constructed in Definition 14 are identical except for the additional nodes $\overline{R} = R_{\text{candidate}} \setminus R$ in G'. By showing that the addition of these nodes does not affect the persistence of a graph, we can prove that the persistence of G' with selection R is equal to that of G with placement R. Consider an optimal attack A in G'. First, A cannot contain any edge connected to a node in $\overline{R}$, since these edges are all directed towards nodes in $\overline{R}$ and, therefore, no path leading to a sink can contain any of these edges. Second, the nodes in $\overline{R}$ do not affect the overall weight of nodes separated by any attack as their weights are all set to zero. Therefore, the set of optimal attacks and the ratios of overall costs to overall separated weights for these attacks are the same for the two graphs. □

Since $\forall_{v \in \mathcal{V}(G)}\ c(v) = \infty$, any selection in G' including a $v \notin R_{\text{candidate}}$ has infinite cost. Therefore, if there is a feasible placement for G then $\mathcal{A}$ always selects an $R \subseteq R_{\text{candidate}}$. As $\mathcal{A}$ selects a minimum set of nodes, R_{opt} is an optimal solution to the problem of placement with required persistence π_0 in G constrained such that nodes can only be placed at $R_{\text{candidate}}$.

Corollary 2. *The above algorithm finds an optimal solution to the problem of sink placement with required persistence.*

The claim of this corollary readily follows from the above and Theorem 15.

The choice of the algorithm $\mathcal{A}$ for sink selection is completely arbitrary. If the goal is to find an approximate solution only, then even polynomial-time algorithms can be used. In this case, as the number of candidate locations and the time needed to enumerate them are also polynomial, the total running time of the algorithm is polynomial as well.

3.6.3 Other Applications of the Proposed Search Space Reduction Technique

The proposed technique can also be applied to problems other than placement with required persistence. In fact, it can be used for any problem where sinks with fixed radii have to be placed so that a measure based only on the topology of the network is maximized.

In the following, I list a few examples where the proposed technique could be used as an improvement:

- In [Poe and Schmitt, 2007], a similar reduction technique is proposed, which determines the set of candidate locations by sampling all possible intersection regions of the sensor nodes' transmission ranges. As the performance measure to be optimized depends only on the topology of the network, it does not matter which location is selected from a given region and, therefore, the technique guarantees an optimal solution. However, it also enumerates points from "inferior regions", which cover only strict subsets of nodes covered by points in some neighboring regions. As the proposed technique does not enumerate such locations, it produces a smaller candidate set.

- In [Youssef and Younis, 2007] and [Youssef and Younis, 2010], the sensor nodes are first clustered using a genetic algorithm and then a sink node is placed for each cluster. For a given cluster, the area around its centroid is divided into a two dimensional grid. For each grid cell, the number of sensors whose transmission range covers the sink is determined and the cell with the maximum count value is selected. Clearly, there is no guarantee that the grid contains an optimal location. Therefore, searching the set of candidate locations determined using the proposed technique instead would be an improvement.

- In [Capone et al., 2010] and [Gandham et al., 2003], the set of candidate locations is assumed to be given. As the used performance measures depend only on topology, the proposed technique could be used to determine the set of candidate locations.

Unfortunately, if the distances between the nodes and the sinks covering them are also taken into consideration, then it is not guaranteed that the subsets of the candidate locations contain an optimal solution. Consequently, the proposed technique cannot be used for problems based on minimizing link lengths.

3.7 Numerical Results

In this section, I present numerical results on the algorithms and techniques proposed in the preceding sections. In the first experiment, I have evaluated the practical performance of the proposed heuristic algorithms for sink selection. In the second experiment, I have measured the average number of candidate locations determined by the proposed search space reduction technique in order to demonstrate that it is sufficiently low for the proposed technique to be applicable in practice. In the third experiment, I have compared the proposed search space reduction technique to others used in the literature, based on the average cost of the best possible placement for of each technique.

3.7.1 Comparison of the Proposed Sink Selection Algorithms

I have studied two performance measures: (1) the ratios between the total selection costs of the sinks in the case of the heuristic algorithms and in the case of the optimal solution and (2) the running times of the heuristic algorithms and an integer programming solver.

In order to obtain reliable values, a large number of networks were generated in a probabilistic manner. The most prevalent model of a wireless sensor network is a unit disk graph, which models a wireless network where each node has the same transmission radius, and two nodes are considered to be neighbors if they are within each other's transmission range. In the simulations, I generated graphs of this type in a probabilistic manner. More precisely, a given number of nodes were placed uniformly at random on a disk of unit radius, and the transmission radius of the nodes was calculated from a given expected average node degree using the approximations given in [Bettstetter, 2004]. Disconnected graphs were connected using minimum distance extra edges.

Edge attack costs, node values and node selection costs were drawn from uniform distributions on $[0.5, 1.5]$, while the required persistence and the expected average node degree were set to 1 and 4, respectively. The cost ratios were computed for each randomly generated graph, and the arithmetic mean of the ratios was taken as the average cost ratio. The experiments have been run for various numbers of nodes, ranging from 16 to 32.

Figure 3.3 shows the ratio of the selection cost of heuristic algorithms to the selection cost of the optimal solution as a function of the node count. Different curves belong to different heuristic algorithms, namely, to the proposed greedy and genetic algorithms. As the optimal solution minimizes the selection cost, the cost selection ratio shown in the figure cannot be smaller than 1, and the closer it is to 1, the better the performance of the heuristic solution is.

In the case of the greedy algorithm, the excess requirement in selection cost fluctuates around 20% and the performance seems to be quite stable with respect to the number of nodes. The performance of the genetic algorithm is almost optimal if the number of nodes is low, and still better than that of the greedy algorithm for higher node counts. In the case of even higher node counts, i.e., node counts larger than 32, the performance of the genetic algorithm is only slightly better than that of the greedy algorithm. As finding optimal solutions is very hard, I have not plotted the cost ratios in this case.

Figure 3.4 shows the average running times of the greedy algorithm, the genetic algorithm, and the integer programming solver as a function of the node count, measured on an average desktop PC. For solving integer programs, `lp_solve`[4], a free, open source mixed integer linear programming solver was used, which is based on the Branch-and-Bound method combined with the revised simplex method.

As expected, the running time of the integer programming solver is exponential and grows faster than those of the heuristic algorithms by several orders of magnitude. Of the two proposed heuristics, the genetic algorithm is faster than the greedy algorithm by an order of magnitude. For node counts higher than 32, the difference between the performance results of the heuristic algorithms is more prominent. For example, for 64 nodes, the running time of the greedy algorithm is 6834 ms, while that of the genetic algorithm is only 452 ms.

[4] lpsolve.sourceforge.net

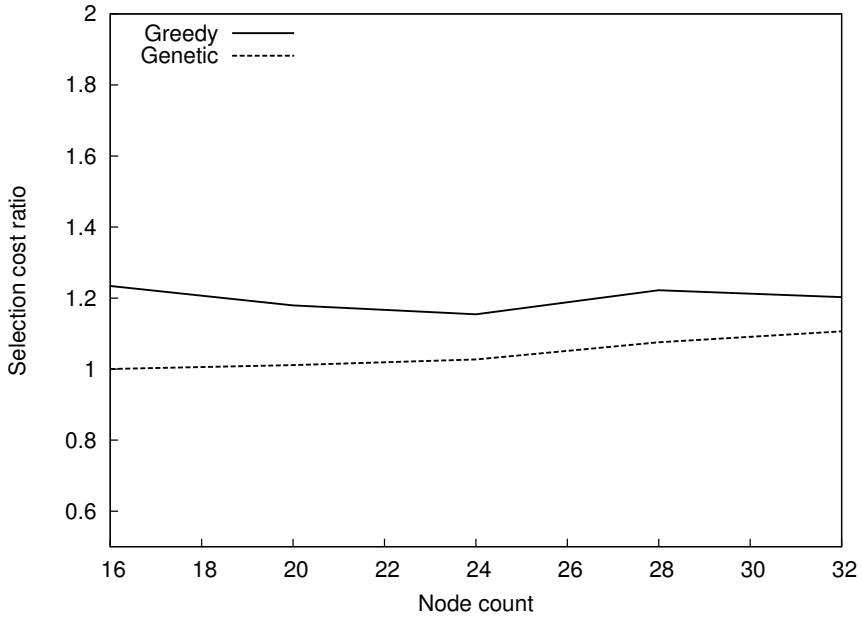


Figure 3.3: Ratios between the cost of the sink selection in the case of a heuristic algorithm and in the case of the optimal solution for different heuristic algorithms and node counts.

3.7.2 Performance of the Proposed Search Space Reduction Technique

Again, the networks on which I have conducted the measurements were generated in a probabilistic manner. Nodes were randomly placed on a disk of unit radius according to a uniform distribution. The experiment has been run for various numbers of nodes and sink radii, the former ranging from 100 to 500.

Figure 3.5 shows the average number of candidate locations. As the exact value of the sink radius is not very informative, the expected average number of nodes on a disk of the given radius (i.e., the average number of nodes covered by a randomly placed sink) is displayed instead. As expected, the number of candidate locations grows linearly with the number of nodes and the rate of the growth is determined by the radii of the sinks. Furthermore, the two numbers are roughly of the same order of magnitude. Therefore, the sink placement problem is practically only as hard as the sink selection problem. Surprisingly, the relationship between the number of nodes in a sink's radius and the number of candidate locations also seems to be linear.

For every combination of parameters presented above, the running time of the proposed algorithm for enumerating candidate locations was in the order of minutes on an average desktop PC. Similarly to the number of candidate locations, the running time of the enumeration also grows linearly with the size of the network. Therefore, we can say that the running time needed to enumerate candidate locations is never going to be a bottleneck, even for large networks. For this reason, I omit the exact numerical results on these running times.

3.7.3 Comparison of Different Search Space Reduction Techniques

The networks on which I have conducted the measurements were generated in a probabilistic manner similar to the one in Subsection 3.7.1. Edge attack costs and node values were drawn from uniform distributions on $[0.05, 0.15]$ and $[0.5, 1.5]$, respectively. The required persistence was set to 0.1. The transmission radii of regular nodes and sink nodes were chosen such that the expected average numbers of nodes on disks of the given radii were 4 and 8, respectively. The experiment has been run for various numbers of nodes, ranging from 16 to 32.

I have compared four different search space reduction techniques:

- Uniform grid: The area where sinks are to be placed is divided into a regular two dimensional grid and the position of each gridpoint is added to the set of candidate locations. To achieve a fair

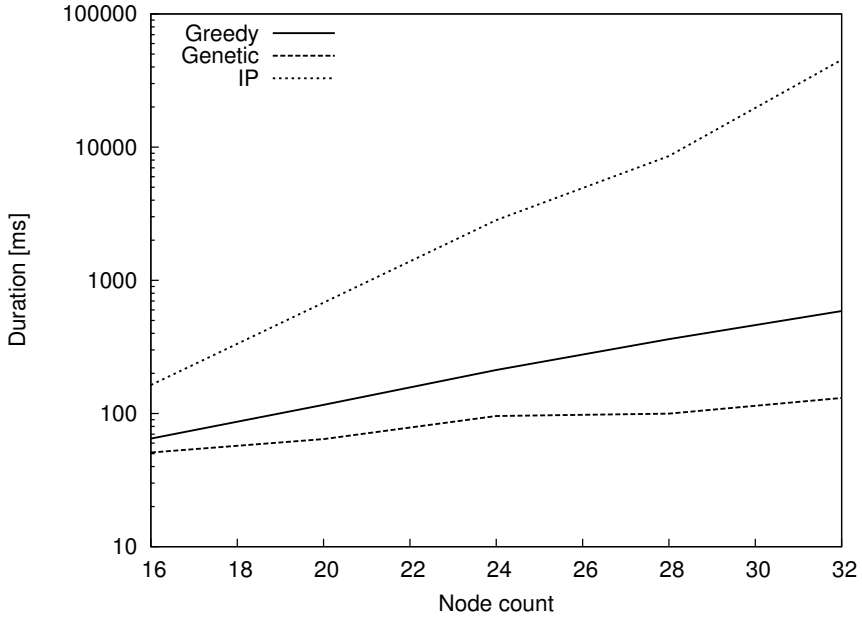


Figure 3.4: Average running time of the greedy algorithm, the genetic algorithm and the integer programming based optimal solution for various node counts. Please, note the logarithmic scale on the y axis.

comparison, the granularity of the grid was chosen such that the number of candidate locations was roughly equal to that of the other techniques.

- "Selection": The set of candidate locations consists of the positions of regular nodes. This technique corresponds to the case when the sink placement problem degenerates to the sink selection problem, hence the name.
- Random: Candidate locations are chosen uniformly at random from the area where sinks are to be placed. To achieve a fair comparison, the number of candidate locations was roughly equal to that of the other techniques.
- Optimal: The set of candidate locations is determined using the proposed technique.

For each network, a set of candidate locations was determined using each search space reduction technique. Then, a minimum cost sink placement with the required persistence was found using an optimal algorithm for sink selection, as it has been described in Subsection 3.6.2. Finally, for each search space reduction technique, the average cost of the optimal placements was calculated, where the average was taken over all generated networks.

Figure 3.6 shows the average cost of the best possible placement as a function of the node count. The different curves belong to the different search space reduction techniques. As expected, the proposed technique clearly outperforms the other three in terms of the average costs of placements.

3.8 Related Work

The optimal placement of sink nodes in wireless sensor networks is an important problem, which has been extensively studied in the literature. For a comprehensive survey on sink node placement, see [Akkaya et al., 2007]. For a general survey on node placement in wireless sensor networks, see [Younis and Akkaya, 2008].

In *single sink placement* problems, the network contains only a single sink node, which is often referred to as the base station. The goal is to find a sink location for which a performance metric is maximal. The most commonly used metrics include the lifetime of the network measured as the time until the most loaded node [Arkin et al., 2010, Pan et al., 2005] or until a given fraction of the nodes run out of

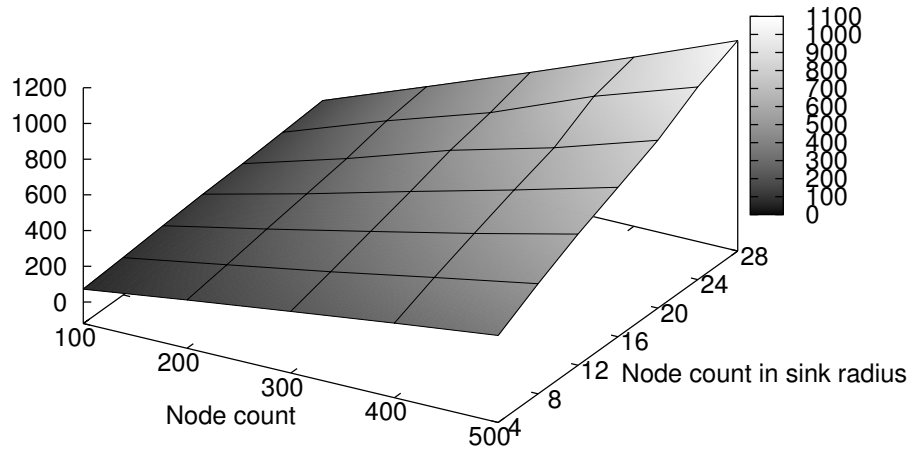


Figure 3.5: Average number of candidate locations for various node counts and sink radii.

battery [Pan et al., 2005], the number of sensors that can transmit their data [Arkin et al., 2010], and the maximum throughput [Muthaiah and Rosenberg, 2008].

In *multiple sink placement* problems, the network can contain more than one sink nodes. These nodes are often referred to as gateways, because they forward the collected data to a center on longer range links. If the number of sinks that can be placed is constrained, the goal is to find a set of locations for which a performance metric is maximal. The most commonly used metrics include the worst case delay measured in the maximum number of hops between any node and the nearest sink [Poe and Schmitt, 2007, Wong et al., 2004], the total number of hops between each node and the nearest sink [Youssef and Younis, 2007, Wong et al., 2004], the network lifetime measured as the time until the most loaded node runs out of battery [Capone et al., 2010, Shi et al., 2006] or until a given fraction of nodes become unreachable [Oyman and Ersoy, 2004], and the network capacity or data rate [Shi et al., 2006, Bogdanov et al., 2004].

If the number of sink nodes is not known in advance, the problem includes finding the minimum number of sinks that is feasible for a given constraint. The most commonly used constraints include the lifetime of the network measured as the time until a fraction of nodes become unreachable [Oyman and Ersoy, 2004] or until the most loaded node dies [Capone et al., 2010], the maximum number of hops between any node and the nearest sink [Youssef and Younis, 2010, Wong et al., 2004], and the total number of hops between each node and the nearest sink [Wong et al., 2004].

The problem can also be defined such that the number of sinks have to be minimized and a given performance metric has to be optimized simultaneously. In this case, the goal can be to find a set of Pareto optimal solutions [Czajko and Wojciechowski, 2010], or to find the optimal solution for a given trade-off ratio between the number of sinks and the performance metric [Capone et al., 2010].

A prevalent approach to solving multiple sink placement problems is sensor node *clustering*. In a clustering scheme, the nodes of the network are first grouped into disjoint clusters, which will be served by distinct sink nodes. Then, a sink node, which in this case is usually called a cluster-head, can be placed for each cluster using a single sink placement algorithm. While the assignment of sensor nodes to sinks after sink placement is sometimes also called clustering, here I only use the term clustering in the above sense. Naturally, the number of clusters, and therefore, the number of sinks can either be part of the objective function or be a pre-specified number. In the former case, the number of clusters has to be minimized under some constraints. For example, in [Aoun and Boutaba, 2006], the size and radius of each cluster and the maximum amount of traffic each node has to relay are limited. In the latter case, the nodes have to be assigned to a fixed number of clusters. Probably the most widespread method used to achieve this is k-means clustering, which, for example, is used by the GOALE algorithm proposed in [Youssef and Younis, 2010]. If multi-level networks are allowed, hierarchical clustering algorithms can also be used. For example, in [Bandyopadhyay and Coyle, 2003], cluster-heads aggregate data from cluster members and forward it to the next level cluster-head. A survey on clustering algorithms for wireless sensor networks can be found in [Abbasi and Younis, 2007].

One of the challenges of sink placement is the infinite size of the search space; therefore, the problem

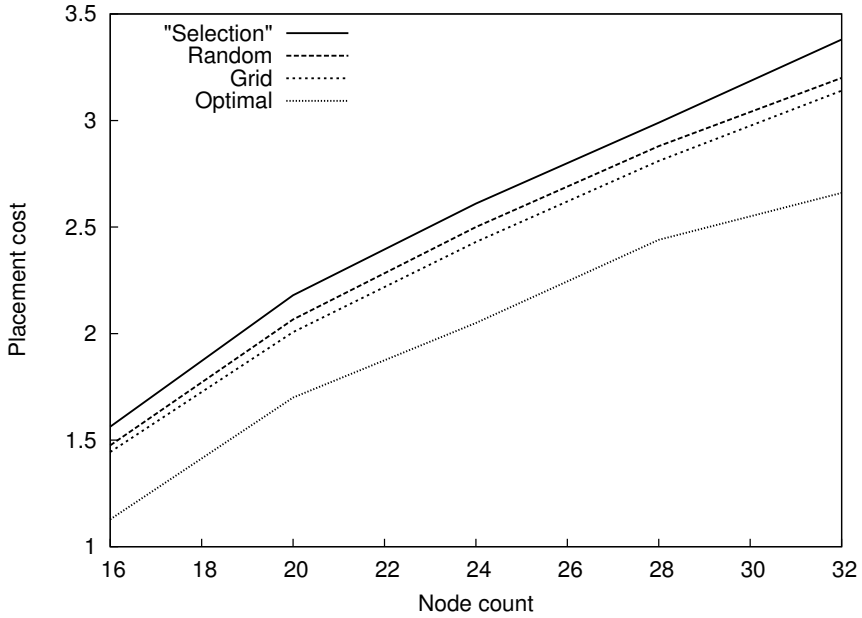


Figure 3.6: The average costs of best possible sink placements for different search space reduction techniques and node counts.

is often constrained by restricting the possible positions of the sinks to a set of candidate locations. If sinks can only be placed at the locations of the sensor nodes, such as in [Aoun and Boutaba, 2006, Bandyopadhyay and Coyle, 2003], the problem is reduced to selecting a subset of nodes to be sinks, which I will refer to as a *sink selection* problem. The set of candidate locations can also be determined by an algorithm based on the geometry of sensor node positions. A wide variety of such algorithms have been proposed, for example, finding the intersections of circles centered at the nodes [Arkin et al., 2010], sampling locations from the intersecting regions of disks centered at the nodes [Poe and Schmitt, 2007], creating two dimensional grids centered at the centroids of clusters after the nodes have been clustered [Youssef and Younis, 2010], or sampling locations from dominating intersecting regions [Wong et al., 2004]. The set of candidate locations can also be pre-specified, i.e., it can be an input parameter of the problem. For example, this set is an arbitrary set in [Capone et al., 2010, Czajko and Wojciechowski, 2010], whereas in [Muthaiah and Rosenberg, 2008], a certain number of pre-specified locations are given from which one has to be selected as the location of the sink, while the others become the locations of the sensor nodes.

Since most sink placement problems are NP-hard, even when they are constrained to sink selection, heuristic, metaheuristic and approximation algorithms are used to solve them in practice. Greedy algorithms [Aoun and Boutaba, 2006, Bogdanov et al., 2004] and other heuristics employing greedy decisions [Capone et al., 2010, Wong et al., 2004] are probably the most prevalent heuristic approaches. Among metaheuristics, genetic algorithms are the most often used [Poe and Schmitt, 2007, Youssef and Younis, 2007, Youssef and Younis, 2010] as they are likely to produce good solutions, even though they have no performance guarantees. Besides genetic algorithms, other, more problem specific metaheuristics can also be used [Czajko and Wojciechowski, 2010]. In [Shi et al., 2006], a set of procedures is proposed to design approximation algorithms for sink placement problems under any desired small error bound. Two examples are given, where this framework can be employed, placement to maximize network lifetime and placement to maximize network capacity.

What I have discussed so far is static positioning. If the nature of the application allows the sinks to be relocated and the network to be reconfigured after it has been deployed, then dynamic placement can be used to improve network performance; for example, in [Gandham et al., 2003], sinks are periodically relocated to prolong network lifetime. Node failures can be anticipated by employing reactive reconfiguration schemes; for example, in [Gupta and Younis, 2003], an efficient re-clustering mechanism is proposed to recover sensors from failed clusters.

The main difference between prior results and the results presented in this chapter is that I optimize

for robustness against adversarial attacks measured in persistence (or deployment cost under a persistence bound), which none of the prior work did. The similarity is that I use heuristics as the problem is hard.

3.9 Conclusions

In this chapter, I have addressed the problem of deploying sink nodes in a wireless sensor network such that the resulting network topology be robust against denial-of-service type attacks, such as node destruction, battery exhaustion, and jamming of wireless links. Instead of the more usual approach of using connectivity for quantifying the robustness of network topologies, I used the notion of persistence. I explained and gave illustrative examples of why persistence is a better robustness metric for wireless sensor networks than connectivity. These arguments apply to a wider range of networks, including most types of access networks.

Using persistence as the robustness metric, I have formalized and studied two variants of the sink deployment problem. In the first variant, which I called the sink selection problem, I restricted the set of possible sink locations to the set of sensor node locations. In the second variant, which I called the sink placement problem, I removed this restriction, and allowed sinks to be placed anywhere in the deployment area. In both variants, I aimed at achieving a given level of persistence while minimizing the deployment cost, which is equally hard computationally as maximizing persistence under a given upper bound on the deployment budget. I have proved that both sink selection with required persistence and sink placement with required persistence are NP-hard.

To solve the sink selection problem efficiently, I have proposed greedy and genetic heuristic algorithms. I have shown how the infinite search space of possible placements can be reduced to a set of candidate locations, which is of polynomial size, such that the resulting set always contains an optimal solution. The proposed search space reduction technique may be of independent interest. I have also shown how any sink selection algorithm, including the proposed heuristic algorithms, can be used to find a solution in the reduced search space.

Finally, I have provided experimental results on the performance of the heuristic algorithms for sink selection and the proposed search space reduction technique. The results show that the proposed technique could be used to efficiently solve other problems or to enhance the performance of previously proposed algorithms.

3.10 Related Publications

- Laszka, A., Buttyán, L., and Szeszlér, D. (2011). Optimal selection of sink nodes in wireless sensor networks in adversarial environments. In *Proceedings of the 2nd IEEE International Workshop on Data Security and PrivAcy in wireless Networks (D-SPAN)*, pages 1–6
- Laszka, A., Buttyán, L., and Szeszlér, D. (2013a). Designing robust network topologies for wireless sensor networks in adversarial environments. *Pervasive and Mobile Computing*, 9(4):546–563

Chapter 4

Mitigating Covert Compromises

4.1 Introduction

Nowadays, organizations and companies increasingly depend on computing resources for their operation due to the advantages of information technology. However, as a side effect, they are also increasingly threatened by security compromises, which may harm their financial bottomline or adversely affect their reputation. Organizations can implement various security measures to prevent compromises, which typically include technologies to detect known attack vectors. However, recent studies of anti-malware and anti-virus tools have demonstrated their ineffectiveness against novel attack approaches and even incrementally modified known malware.

At the same time, attackers prey upon opportunities to keep successful security compromises covert. The goal is to benefit from defenders' lack of awareness by exploiting resources, and extracting credentials and company secrets for as long as possible. In contrast to non-covert attacks and compromises that focus on short-term benefits, these long-lasting and (for typical organizations) undetectable attacks pose specific challenges to system administrators and creators of security policies. Discoveries of such attacks by sophisticated security companies provide evidence for damage caused over many months or years.

CDorked, a highly advanced and stealthy backdoor, was discovered in April 2013 [ESET Press Center, 2013]. The malware uses compromised webservers to infect visitors with common system configurations. To stay covert, the malware uses a number of different techniques, for example, not delivering malicious content if the visitor's IP address is in a customized blacklist. The operation has been active since at least December 2012, and has infected more than 400 webservers, including 50 from Alexa's top 100,000 most popular websites. As a similar example, NotCompatible, an Android trojan, was also spread by originally non-malicious, but compromised websites [Poeter, 2012].

Cyber-espionage also often relies on keeping compromises covert. A recent example is Gauss, a complex, nation-state sponsored cyber-espionage toolkit, which is closely related to the notorious Stuxnet [Bencsath et al., 2012, Kaspersky Lab, 2012]. Gauss was designed to steal sensitive financial data from targets primarily located in the Middle East, and was active for at least 10 months before it was discovered.

Such recently-revealed attack vectors as well as the suspected number of unknown attacks highlight the importance of developing mitigation strategies to minimize the resulting expected losses. Potentially effective mitigation approaches include resetting of passwords, changing cryptographic private keys, reinstalling servers, or reinstantiating virtual servers. These approaches are often effective at resetting the resource to a known safe state, but they reveal little about past compromises. For example, if a server is reinstalled, knowledge of if and when a compromise occured may be lost. Likewise, resetting a password does not reveal any information about the confidentiality of previous passwords.

Covert (and non-covert) attacks can be distinguished in another dimension by the extent to which the attack is targeted (or customized) for a particular organization [Casey, 2003, Herley, 2010]. Approaches related to cyber-espionage are important examples of targeted attacks, and require a high effort level customized to a specific target. A typical example of a non-targeted covert attack is the recruitment of a computer into a botnet via drive-by-download. Such attacks are relatively low effort, and do not

require a specific target. Further, they can often be scaled to affect many users for marginal additional cost [Herley, 2010]. See Table 4.1 for a comparison between targeted and non-targeted attacks.

Table 4.1: Comparison of Targeted and Non-Targeted Attacks

	Targeted	Non-Targeted
Number of attackers	low	high
Number of targets	low	high
Effort required for each attack	high	low
Success probability of each attack	high	low

The targeted nature of an attack also matters to the defender, because targeted and non-targeted attacks do different types of damage. For example, a targeting attacker might use a compromised computer system to access an organization's secret e-mails, which may potentially cause enormous economic damage; while a non-targeting attacker might use the same compromised machine to send out spam, causing different types of damage.

The presence of both targeted and non-targeted covert attacks presents an interesting dilemma for a medium-profile target to choose a mitigation strategy against covert attacks. Strategies which are optimal against non-targeted attacks may not be the best choice against targeted attacks. At the same time, mitigation strategies against targeted attacks may not be economically cost-effective against only non-targeting attackers.

To address this dichotomy, I present a game in which a defender must vie for a contested resource that is subject to both targeted attacks from a strategic attacker, and non-targeted covert attacks from a large set of non-strategic attackers. I identify Nash equilibria in the simultaneous game, and subgame-perfect equilibria in the sequential game with the defender leading. The optimal mitigation strategies for the defender against these combined attacks lend insights to policy makers regarding renewal requirements for passwords and cryptographic keys.

The outline of this chapter is the following. First, in Section 4.1, I introduce the game-theoretic model on which the analysis of this chapter is based. Then, in Section 4.3, I give analytical results for this model. In Section 4.4, I present numerical and graphical observations. Finally, in Section 4.5, I review related work, and in Section 4.6, I conclude the chapter.

4.2 Game-Theoretic Model

I model the covert compromise scenario as a randomized, one-shot, non-zero-sum game. For a list of the symbols used in the model, see Table 4.2. The player who represents the rightful owner of the resource is called the defender, while the other players are called attackers. The game starts at time $t = 0$ with the resource being uncompromised, and it is played indefinitely as $t \to \infty$. Time is assumed to be continuous.

Table 4.2: List of Symbols

C_D	move cost for the defender
C_A	move cost for the targeting attacker
B_A	benefit received per unit of time for the targeting attacker
B_N	benefit received per unit of time for the non-targeting attackers
F_A	cumulative distribution function of the attack time for the targeting attacker
λ_N	rate of the non-targeted attacks' arrival

Let D, A, and N denote the defender, the targeting attacker, and the non-targeting attackers, respectively. At any time instance, player i may make a move, which costs her C_i (note that, for attackers, I will use the words attack and move synonymously). When the defender makes a move, the resource becomes uncompromised immediately for every attacker. When the targeting attacker makes a move, she starts her attack, which takes some random amount of time. If the defender makes a move

while an attack is in progress, the attack fails. The amount of time required by a targeted attack to succeed is assumed to follow the same distribution every time. This attack time distribution's cumulative function is denoted by F_A. I assume that $F_A(0) = 0$; in other words, I assume that every targeted attack requires some non-zero amount of time. In practice, this distribution can be based on industry-wide beliefs, statistics of previous attacks, etc.

The attackers' moves are stealthy; i.e., the defender does not know when the resource became compromised or if it is compromised at all. On the other hand, the defender's moves are non-stealthy. In other words, the attackers learn immediately when the defender has made a move.

The cost rate for player i up to time t, denoted by $c_i(t)$, is the number of moves per unit of time made by player i up to time t, multiplied by the cost per move C_i.

For attacker $i \in \{A, N\}$, the benefit rate $b_i(t)$ up to time t is the fraction of time up to t that the resource has been compromised by i, multiplied by B_i. Note that, if multiple attackers have compromised the resource, they all receive benefits until the defender's next move. For the defender D, the benefit rate $b_D(t)$ up to time t is $-\sum_{i\in\{A,N\}} b_i(t)$ (in other words, her loss is $\sum_{i\in\{A,N\}} b_i(t)$). Notice that the game would be zero-sum if we only considered the players' benefits.[1] Because the players' payoffs also consider move costs, the game is *not* zero-sum. Player i's payoff is defined as

$$\liminf_{t\to\infty} \; b_i(t) - c_i(t) \; . \tag{4.1}$$

It is important to note that the asymptotic benefit rate $\liminf_{t\to\infty} b_i(t)$ of attacker i is equal to the asymptotic probability that i has the resource compromised at a random time instance, multiplied by B_i. For a discussion on computing the players' payoffs for the most important strategy profiles, see Section 4.2.3.

4.2.1 Types of Strategies for the Defender and the Targeting Attacker

Not Moving

A player can choose to *never move*. While this might seem counter-intuitive, it is actually a best-response if the expected benefit from making a move is always less than the cost of moving.

Adaptive Strategies for the Targeting Attacker

Let $\mathcal{T}(n) = \{T_0, T_1, \ldots, T_n\}$ denote the move times of the defender up to her nth move (or in the case of $T_0 = 0$, the start of the game). The attacker uses an *adaptive strategy* if she waits for $W(\mathcal{T}(n))$ time until making a move after the defender's nth move (or after the start of the game), where W is a non-deterministic function. If the defender makes her $n+1$st move before the chosen wait time is up, the attacker chooses a new wait time $W(\mathcal{T}(n+1))$, which also considers the new information that is the defender's $n+1$st move time. This class is a simple representation of all the rational strategies available to an attacker, since the function W can depend on all the information that the attacker has, and we do not have any constraints on W.

Renewal Strategies

Player i uses a *renewal strategy* if the time intervals between consecutive moves are identically distributed independent random variables, whose distribution is given by the cumulative function F_{R_i}. Renewal strategies are well-motivated by the fact that the defender is playing blindly. Thus, she has the same information available after each move. So it makes sense to use a strategy which always chooses the time until her next flip according to the same distribution Note that renewal strategies are a special subset of adaptive strategies.

[1] Note that I assume that the defender's loss due to compromise is equal to the attackers' benefit only for notational simplicity. Since the best response strategies and – consequently – the equilibrium profiles remain the same if we apply an element-wise positive-affine transformation to the players' payoffs, my analysis actually requires only that the losses and benefits are both proportional to time.

Periodic Strategies

Player i uses a *periodic strategy* if the time intervals between her consecutive moves are identical. This interval, called the period, is denoted by δ_i. Note that periodic strategies are a special subset of renewal strategies.

4.2.2 Non-Targeted Attacks

Suppose that there are N non-targeting attackers. In practice, N is very large, but the expected number of successful compromises in any given time interval is finite. Hence, as N goes to infinity, the probability that a given non-targeting attacker targets the defender approaches zero. Since non-targeting attackers operate independently of each other, the number of successful attacks in any time interval depends solely on the length of the interval. Therefore, the arrival of *successful non-targeted attacks* can be modeled using a *Poisson process*.

Furthermore, since the economic decisions of the non-targeting attackers depend on a very large pool of possible targets, the effect of the defender's behavior on the non-targeting attackers' strategies is negligible. Thus, the non-targeting attackers' strategies (that is, the arrival rate of the Poisson process) can be considered exogenously given. Let λ_N denote the expected number of arrivals that occur per unit of time; and let all the non-targeting attackers represented together as a single attacker, whose benefit per unit of time is B_N.

4.2.3 Payoffs

Here, I derive formulas for the defender's and targeting attacker's payoffs for certain strategy profiles. While the selection of these strategy profiles might seem arbitrary for now, we will see in Section 4.3 that they are of key importance.

Defender Not Moving, Attacker Moving

If the defender does not move, and the targeted attacker moves once after waiting a finite (possibly zero) amount of time, then both types of attacks will eventually have the resource compromised indefinitely. Thus, the asymptotic probability of the resource being compromised is 1 for both types of attacks. Consequently, the defender's payoff is

$$-B_A - B_N \ , \tag{4.2}$$

while the targeted attacker's payoff is

$$B_A \ . \tag{4.3}$$

Note that the targeted attacker bears zero moving cost because she moves once in an infinite amount of time.

Defender Moving Periodically, Attacker Not Moving

Let δ_D denote the period of the defender's strategy. Then, the expected amount of time that the non-targeted attackers have the resource compromised in one period is

$$\int_{a=0}^{\delta_D} (\delta_D - a)\lambda_N e^{-\lambda_N a} \, da \ . \tag{4.4}$$

The first factor, $(\delta_D - a)$, is the amount of time that the resource will be compromised if the attack arrives a time after the start of the period, and the second factor, $\lambda_N e^{-\lambda_N a}$, is the probability density of the attack arriving after a time. We have that the defender loses B_N per unit of time the resource is compromised. Thus, her average loss per unit of time due to non-targeted attacks is

$$\frac{B_N \int_{a=0}^{\delta_D} (\delta_D - a)\lambda_N e^{-\lambda_N a} \, da}{\delta_D} \ . \tag{4.5}$$

By also taking her moving costs into account, we have that the defender's payoff is

$$\frac{B_N \int_{a=0}^{\delta_D} (\delta_D - a)\lambda_N e^{-\lambda_N a} \, da - C_D}{\delta_D} \ , \tag{4.6}$$

while the targeted attacker's payoff is obviously 0.

Defender Moving Periodically, Attacker Moving Immediately

From the previous case, we already have a formula for the defender's losses due to non-targeted attacks. Using the same argument, we have that the expected amount of time that the targeted attacker has the resource compromised in a single period is

$$\int_{a=0}^{\delta_D} (\delta_D - a) f_A(a)\ da\ , \tag{4.7}$$

and the average loss per unit of time due to targeted attacks is

$$\frac{B_A \int_{a=0}^{\delta_D} (\delta_D - a) f_A(a)\ da}{\delta_D}\ . \tag{4.8}$$

By also taking the move cost (C_D for the defender and C_A for the attacker) for each period into account, we have that the defender's payoff is

$$\frac{-B_A \int_{a=0}^{\delta_D} (\delta_D - a) f_A(a)\ da - B_N \int_{a=0}^{\delta_D} (\delta_D - a) \lambda_N e^{-\lambda_N a}\ da - C_D}{\delta_D} \tag{4.9}$$

$$= \frac{-\int_{a=0}^{\delta_D} (\delta_D - a) B_A f_A(a) + B_N \lambda_N e^{-\lambda_N a}\ da - C_D}{\delta_D}\ , \tag{4.10}$$

and the targeted attacker's payoff is

$$\frac{B_A \int_{a=0}^{\delta_D} (\delta_D - a) f_A(a)\ da - C_A}{\delta_D}\ . \tag{4.11}$$

4.2.4 Comparison to `FlipIt`

Even though this game-theoretic model resembles `FlipIt` (see the related work in Section 4.5) in many ways, it differs in three key assumptions.

- First, the defender's moves are assumed to be *non-stealthy*. The motivation for this is that an attacker should know whether she has the resource compromised or not if she receives benefits from continuously exploiting the compromised resource. For example, if the attacker uses the compromised password of an account to regularly spy on its e-mails, she will learn of a password reset immediately when she tries to access the account.
- Second, the targeting attacker's moves are assumed to be *non-instantaneous*. The motivation for this is that every targeted attack requires some amount of time and effort to be carried out in practice. Furthermore, the time required for a successful attack may vary, which is modeled by representing the attack time as a random variable.
- Third, the defender is assumed to face *multiple attackers*, not only a single one.

Moreover, to my best knowledge, papers published on `FlipIt` so far give analytical results only on a very restricted set of strategies. In contrast, I completely describe the game's equilibria and give optimal defender strategies based on very mild assumptions, which effectively do not limit the power of players (see the introduction of Section 4.3).

4.3 Analytical Results

In this section, I provide analytical results on the game. I start with a discussion on the players' strategies.

First, recall that the defender has to play blindly, which means that she has the same information available after each one of her moves. Consequently, it makes sense for her to choose the time until her next flip according to the same distribution each time. In other words, a rational defender can restrict herself to using only renewal strategies.

Now, if the defender uses a renewal strategy, the time of her next move depends only on the time elapsed since her last move T_n, and the times of her previous moves (including T_n) are irrelevant to the future of the game. Therefore, it is reasonable to assume that the attacker's response strategy to a renewal strategy also does not depend on $T_0, T_1, \ldots, T_n$. For the remainder of the paper, when the defender plays a renewal strategy, the attacker uses a fixed probability distribution – given by the density function f_W – over her wait times for when to begin her attack. Note that it is clear that there always exists a best-response strategy of this form for the attacker against a renewal strategy.

Since the attacker always waits an amount of time that is chosen according to a fixed probability distribution after the defender's each move, the amount of time until the resource would be successfully compromised after the defender's move also follows a fixed probability distribution. Let S be the random variable measuring the time after the defender has moved until the attacker's attack would finish. The probability density function f_S of S can be computed as

$$f_S(s) = \int_{w=0}^{s} f_W(w) \int_{a=0}^{(s-w)} f_A(a)\ da\ dw\ . \tag{4.12}$$

Finally, I use the standard notation F_S to denote the cumulative distribution function of S.

4.3.1 Best Responses

Defender's Best Response

I begin the analysis with finding the defender's best-response strategies. Against a targeting attacker who uses an adaptive strategy, the best response is given by the following lemma.

Lemma 6. *Suppose that the non-targeted attacks arrive according to a Poisson process with rate λ_N, and the targeting attacker uses an adaptive strategy with a fixed wait time distribution F_W. Then,*

- *not moving is the only best response if $C_D = \mathcal{D}(l)$ has no solution for $l > 0$, where*

$$\mathcal{D}(l) = B_A \left(l F_S(l) - \int_{s=0}^{l} F_S(s)\ ds \right) + B_N \left(-l e^{-\lambda_N l} + \frac{1 - e^{-\lambda_N l}}{\lambda_N} \right)\ ; \tag{4.13}$$

- *the periodic strategy whose period is the unique solution to $C_D = \mathcal{D}(l)$ is the only best response otherwise.*

Even though we cannot express the solution of $C_D = \mathcal{D}(l)$ in closed form, it can be easily found using numerical methods, as the right hand side is continuous and increasing.[2] Note that all the equations presented in the subsequent lemmas and theorems of this paper can also be solved easily using numerical methods.

Proof. When playing a renewal strategy, the defender randomly selects the intervals between her consecutive moves according to the distribution generating her strategy. In a best response, her strategy and, hence, every interval length in the support of the generating distribution has to maximize her expected payoff per unit of time. In other words, every interval length has to minimize her expected loss per unit

[2] I show in the proof of the lemma that the right hand side is increasing in l.

of time. The defender's expected loss per unit of time for an interval of length l is

$$\frac{1}{l}\Bigg(\quad B_A \int_{s=0}^{l} (l-s) f_S(s)\ ds + B_N \int_{a=0}^{l} (l-a)\lambda_N e^{-\lambda_N a} da + C_D \Bigg) \tag{4.14}$$

$$= \frac{1}{l}\Bigg(\quad B_A \Bigg([(l-s)F_S(s)]_{s=0}^{l} - \int_{s=0}^{l} (-1) F_S(s)\ ds \Bigg)$$

$$+ B_N \left(\frac{e^{-\lambda_N l} - 1}{\lambda_N} + l \right) + C_D \Bigg) \tag{4.15}$$

$$= \frac{1}{l}\Bigg(\quad B_A \Bigg(F_S(l) \underbrace{(l-l)}_{0} - \underbrace{F_S(0)}_{0}(l-0) + \int_{s=0}^{l} F_S(s)\ ds \Bigg) \tag{4.16}$$

$$+ B_N \left(\frac{e^{-\lambda_N l} - 1}{\lambda_N} + l \right) + C_D \Bigg) \tag{4.17}$$

$$= \frac{1}{l}\Bigg(\quad B_A \int_{s=0}^{l} F_S(s)\ ds + B_N \left(\frac{e^{-\lambda_N l} - 1}{\lambda_N} + l \right) + C_D \Bigg) . \tag{4.18}$$

To find the minimizing interval lengths (if there exists any), I take the derivative of (4.18) and solve it for equality with 0 as follows:

$$0 = \frac{d}{dl}\left[\frac{1}{l}\left(B_A \int_{s=0}^{l} F_S(s)\ ds + B_N \left(\frac{e^{-\lambda_N l} - 1}{\lambda_N} + l \right) + C_D \right) \right] \tag{4.19}$$

$$0 = -\frac{1}{l^2}\left(B_A \left(\int_{s=0}^{l} F_S(s)\ ds - l F_S(l) \right) + B_N \frac{e^{-\lambda_N l}(\lambda_N l - e^{\lambda_N l} + 1)}{\lambda_N} + C_D \right) \tag{4.20}$$

$$C_D = B_A \left(l F_S(l) - \int_{s=0}^{l} F_S(s)\ ds \right) + B_N \left(-l e^{-\lambda_N l} + \frac{1 - e^{-\lambda_N l}}{\lambda_N} \right) . \tag{4.21}$$

Firstly, it is easy to see that the first term of the right hand side of the above equation is a non-decreasing function of l, as F_S is a non-negative, non-decreasing function. Secondly, the second term is strictly increasing, as its derivate is $\lambda_N l e^{-\lambda_N l} > 0$. Thus, the right hand side of Equation (4.21) is strictly increasing as a function of l. Consequently, if there is any solution l^* to the above equation, then it has to be unique. Furthermore, this value of l is a minimizing value for the expected loss per unit of time as the second derivative at l^* is greater than zero:

$$\frac{d}{dl}\left[-\frac{1}{l^2}\left(B_A \left(\int_{s=0}^{l} F_S(s)\ ds - l F_S(l) \right) + B_N \frac{e^{-\lambda_N l}(\lambda_N l - e^{\lambda_N l} + 1)}{\lambda_N} + C_D \right) \right] \tag{4.22}$$

$$= \frac{1}{l^3}\Bigg(\quad B_A \left(2 \int_{s=0}^{l} F_S(s)\ ds - 2 l F_S(l) + l^2 f_S(l) \right)$$

$$+ B_N \frac{e^{-\lambda_N l}(\lambda_N^2 l^2 + 2\lambda_N l - 2e^{\lambda_N l} + 2)}{\lambda_N} + 2C_D \Bigg) . \tag{4.23}$$

We care about the value of this expression when the first derivative is zero. Using this constraint, I

obtain

$$\frac{1}{l^3}\Bigg(\quad B_A \left(2\int_{s=0}^{l} F_S(s)\ ds - 2lF_S(l) + l^2 f_S(l) \right) + B_N \frac{e^{-\lambda_N l}(\lambda_N^2 l^2 + 2\lambda_N l - 2e^{\lambda_N l} + 2)}{\lambda_N} + 2C_D \Bigg) \tag{4.24}$$

$$= \frac{2}{l^3}\Bigg(\underbrace{B_A \left(\int_{s=0}^{l} F_S(s)\ ds - lF_S(l) \right) + B_N \frac{e^{-\lambda_N l}(\lambda_N l - e^{\lambda_N l} + 1)}{\lambda_N} + C_D}_{\text{first derivate } \cdot(-l^2)} \Bigg)$$

$$+ \frac{1}{l}\left(B_A f_S(l) + B_N e^{-\lambda_N l}\lambda_N \right) \tag{4.25}$$

$$= \frac{2}{l^3}\Big(0\Big) + \frac{1}{l}\left(B_A \underbrace{f_S(l)}_{\geq 0} + B_N \underbrace{e^{-\lambda_N l}}_{>0}\lambda_N \right) > 0\ . \tag{4.26}$$

Thus, if Equation (4.21) has a solution for l, then this solution is the only interval length that minimizes the defender's loss. Consequently, her only best response is the periodic strategy with the minimizing l^* as the period.

On the other hand, if Equation (4.21) is not satisfiable for l, then her only best response for the defender is to never move. When $l \to \infty$, the defender's expected loss per unit of time approaches $B_A + B_N$, which is equal to her loss when she never moves. When $l \to 0$, her expected loss per unit of time goes to infinity due to the ever increasing costs. Consequently, if there is no minimizing l, then the defender's expected loss per unit of time is greater than $B_A + B_N$ for every interval length l. □

Against a targeting attacker who never attacks, the best response is given by the following lemma.

Lemma 7. *Suppose that the non-targeted attacks arrive according to a Poisson process with rate λ_N, and the targeting attacker never attacks. Then,*

- *not moving is the only best response if $C_D = \mathcal{D}^N(l)$ has no solution for $l > 0$, where*

$$\mathcal{D}^N(l) = B_N \left(-le^{-\lambda_N l} + \frac{1 - e^{-\lambda_N l}}{\lambda_N} \right)\ ; \tag{4.27}$$

- *the periodic strategy whose period is the unique solution to $C_D = \mathcal{D}^N(l)$ is the only best response otherwise.*

Proof. Follows readily from the proof of Lemma 6 with the terms belonging to the targeting attacker omitted everywhere. □

Observe that $\mathcal{D}(0) = \mathcal{D}^N(0) < 0$ and $\mathcal{D}(l) \geq \mathcal{D}^N(l)$. Consequently, $C_D = \mathcal{D}(l)$ has a solution whenever $C_D = \mathcal{D}^N(l)$ has one. Furthermore, if both have solutions, the solution of $C_D = \mathcal{D}(l)$ is less than or equal to the solution of $C_D = \mathcal{D}^N(l)$. In other words, the *defender is more likely to keep moving if there is a threat of targeted attacks*, and she *will move at least as frequently as she would if there was no targeting attacker.*

Attacker's Best Response

I continue the analysis with finding the attacker's best-response strategy.

Lemma 8. *Against a defender who uses a periodic strategy with period δ_D,*

- *never attacking is the only best response if $C_A > \mathcal{A}(\delta_D)$, where*

$$\mathcal{A}(\delta) = B_A \int_{a=0}^{\delta} F_A(a)da\ ; \tag{4.28}$$

- *attacking immediately after the defender has moved is the only best response if* $C_A < \mathcal{A}(\delta_D)$*;*
- *both not attacking and attacking immediately are best responses otherwise.*

Proof. First, assume that the attacker does attack. Given that the attacker waits $w < \delta_D$ time before making her move, the expected amount of time she has the resource compromised until the defender's next move is

$$\int_{a=0}^{\delta_D - w} f_A(a)(\delta_D - w - a)\ da\ . \tag{4.29}$$

It is easy to see that the maximum of this equation is attained for $w = 0$. Therefore, if the attacker does attack, she attacks immediately. Then, the expected amount of time she has the resource compromised until the defender's next move is

$$\int_{a=0}^{\delta_D} f_A(a)(\delta_D - a)\ da \tag{4.30}$$

$$= [F_A(a)(\delta_D - a)]_{a=0}^{\delta_D} - \int_{a=0}^{\delta_D} F_A(a)(-1)\ da \tag{4.31}$$

$$= F_A(\delta_D)(\underbrace{\delta_D - \delta_D}_{0}) - \underbrace{F_A(0)}_{0}(\delta_D - 0) + \int_{a=0}^{\delta_D} F_A(a)\ da \tag{4.32}$$

$$= \int_{a=0}^{\delta_D} F_A(a)\ da\ . \tag{4.33}$$

Therefore, if the attacker does attack, her asymptotic benefit rate is

$$B_A \frac{\int_{a=0}^{\delta_D} F_A(a) da}{\delta_D}\ , \tag{4.34}$$

and her payoff is

$$B_A \frac{\int_{a=0}^{\delta_D} F_A(a) da}{\delta_D} - \frac{C_A}{\delta_D}\ . \tag{4.35}$$

Thus, when (4.35) is less than or equal to zero, never attacking is a best-response strategy; when (4.35) is greater than or equal to zero, always attacking immediately is a best-response strategy. Finally, when (4.35) is equal to zero, the attacker can decide whether to attack immediately or not to attack at all after each move of the defender, as both choices maximize her payoff. □

The lemma shows that the targeting attacker should either attack immediately or not attack at all, but she should never wait to attack. For the never attack strategy, we already have the defender's best response from Lemma 7. For the attacking immediately strategy, the defender can determine the optimal period of her strategy *solely based on the distribution of* A, which is an exogenous parameter of the game. More formally, the defender's best response is not to move if $C_D = \mathcal{D}^A(l)$ has no solution, and it is a periodic strategy whose period is the unique solution to $C_D = \mathcal{D}^A(l)$ otherwise, where

$$\mathcal{D}^A(l) = B_A \left(l F_A(l) - \int_{a=0}^{l} F_A(a)\ da \right) + B_N \left(-l e^{-\lambda_N l} + \frac{1 - e^{-\lambda_N l}}{\lambda_N} \right)\ . \tag{4.36}$$

This follows readily from Lemma 6 by substituting F_S for F_A.[3]

4.3.2 Nash Equilibria

Based on the previous lemmas, I describe all the equilibria of the game (if there are any) as follows.

Theorem 18. *Suppose that the defender uses a renewal strategy, the targeting attacker uses an adaptive strategy, and the non-targeted attacks arrive according to a Poisson process. Then, the game's equilibria can be described as follows.*

[3]Recall that S was defined as the sum of the waiting time W, which is always zero in this case, and the attack time A.

1. *If $C_D = \mathcal{D}^A(l)$ does not have a solution for l, then there is a unique equilibrium in which the defender does not move and in which the targeting attacker moves once at the beginning of the game.*

2. *If $C_D = \mathcal{D}^A(l)$ does have a solution δ_D for l:*

   (a) *If $C_A \leq \mathcal{A}(\delta_D)$, then there is a unique equilibrium in which the defender plays a periodic strategy with period δ_D, and the targeting attacker moves immediately after the defender's each move.*

   (b) *If $C_A > \mathcal{A}(\delta_D)$,*

      i. *if $C_D = \mathcal{D}^N(l)$ has a solution δ'_D for l, and $C_A \geq \mathcal{A}(\delta'_D)$, then there is a unique equilibrium in which the defender plays a periodic strategy with period δ'_D, and the targeting attacker never moves;*

      ii. *otherwise, there is no equilibrium.*

For an illustration of the hierarchy of the theorem's criteria, see Figure 4.1. Finally, recall that formulas for the player's payoffs can be found in Section 4.2.3.

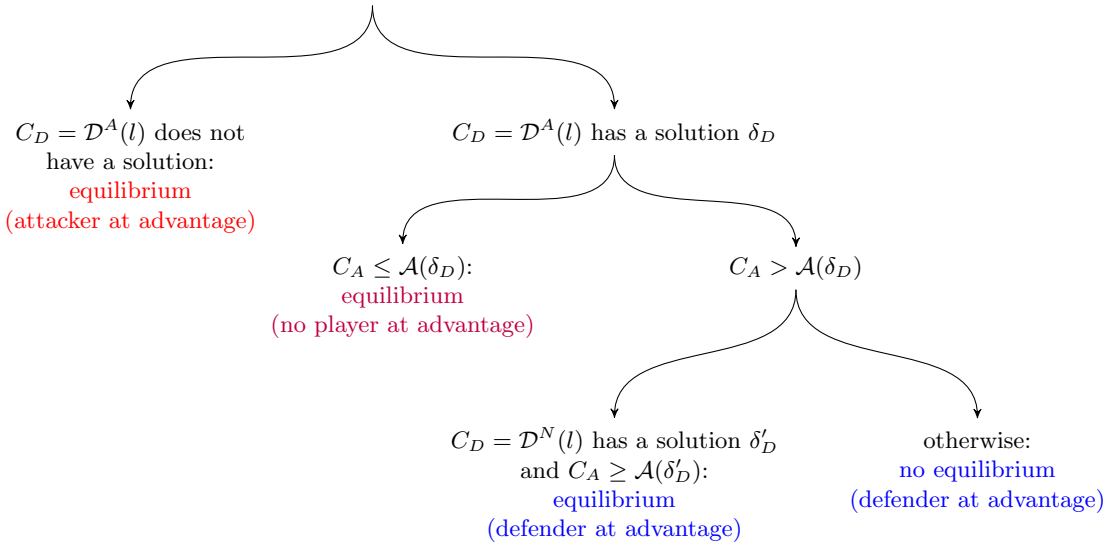
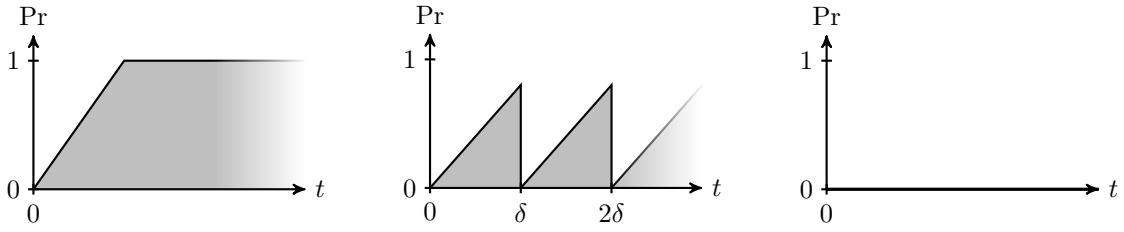


Figure 4.1: Illustration for the hierarchy of criteria in Theorem 18.



(a) Case 1. (b) Case 2. (a) (c) Case 2. (b) i.

Figure 4.2: The probability that the targeting attacker has compromised the resource (vertical axis) as a function of time (horizontal axis) in various equilibria (see Theorem 18 for each case). Note that these are just examples, the actual shapes of the function depend on F_A.

In the first case (Case 1.), the attacker is at an overwhelming advantage, as the relative cost of defending the resource is prohibitively high. Consequently, the defender simply "gives up," as any effort to protect the resource is not profitable, and the attacker will eventually have the resource compromised

indefinitely (see Figure 4.2 for an illustration). In the second case (Case 2. (a)), no player is at an overwhelming advantage. Both players are actively moving, and the resource gets compromised and uncompromised from time to time. In the third and fourth cases (Cases 2. (b) i. and ii.), the defender is at an overwhelming advantage. However, this does not necessarily lead to an equilibrium. If the defender moves with a sufficiently high rate, she makes moving unprofitable for the targeting attacker. But if the targeting attacker decides not to move, then the defender switches to a lower move rate, which is optimal against only non-targeted attacks. However, once the defender switches to the lower move rate, it might again be profitable for the targeting attacker to move, which would in turn trigger the defender to switch back to the higher move rate.

Proof. First, we have from Lemma 6 that, in any equilibrium, the defender either never moves or uses a periodic strategy. If the defender never moves, then the attacker's best response is to wait a finite (possibly zero) amount of time after the game starts, as she will eventually compromise the resource indefinitely. On the other hand, if the defender moves using a periodic strategy, we have from Lemma 8 that the attacker either never attacks or attacks immediately. This leaves us with two strategies for defender (never moving or moving periodically) and three strategies for the targeting attacker (never moving, moving immediately, or waiting a finite non-zero amount of time) from which all equilibria must be composed. Next, I show that only three combinations of these strategies can form equilibria.

It is easy to see that if the defender never moves, the attacker's only best response is to wait a finite (possibly zero) amount of time, since she will eventually compromise the resource indefinitely. Consequently, there is no equilibrium in which both the defender and the targeting attacker are not moving. This leaves us with three possible strategy profiles for the equilibria: the defender moving periodically and the targeting attacker never attacking, the defender moving periodically and the targeting attacker attacking immediately, and the defender never moving and targeting attacker moving after a finite (possibly zero) amount of time.

Now, I prove the theorem case-by-case.

- If $C_D = \mathcal{D}^A(l)$ does not have a solution for l, then neither does $C_D = \mathcal{D}^N(l)$. Thus, the defender's best response is not moving, regardless of whether the targeting attacker uses some short waiting time or never attacks (Lemmas 6 and 7). Note that, however, if the attacker uses some long (but finite) waiting time, moving might be a best response for the defender. Consequently, in order for an equilibrium to exist, the attacker has to use some short waiting time (we know that there is at least one waiting time that is short enough: 0, i.e., attacking immediately). Therefore, the only possible equilibrium is the defender never moving and the targeting attacker moving after a short (possibly) zero amount of time. Since this is a best response for the attacker (Lemma 8), this is a unique equilibrium (I do not distinguish between the short waiting times here, as they lead to the same payoffs and are equivalent in effect).

- If $C_D = \mathcal{D}^A(l)$ does have a solution δ_D for l, then the defender not moving but the targeting attacker moving cannot be an equilibrium, since not moving would not be a best response for the defender (Lemma 6). Thus, for all remaining cases, we are left with two strategy profiles for the equilibria (the defender moving periodically and the attacker never moving or moving immediately).

  Now, if $C_A \leq \mathcal{A}(\delta_D)$, then the attacker's best response to the defender moving periodically with period δ_D is to attack immediately (Lemma 8).[4] Thus, both players moving is an equilibrium. I show that this equilibrium is unique. For the sake of contradiction, assume that this is not true, in other words, there exists an equilibrium in which the defender uses another period $\tilde{\delta}_D$. Clearly, the attacker's strategy in this equilibrium has to be never moving: if the attacker was moving, then the defender's unique best response would be δ_D (Lemma 6). Thus, $C_A > \mathcal{A}(\tilde{\delta}_D)$ has to hold, which implies that $\tilde{\delta}_D < \delta_D$ since $\mathcal{A}(\delta)$ is increasing in δ and $C_A \leq \mathcal{A}(\delta_D)$. But this leads to a contradiction, as the defender's best response to the targeting attacker not moving is a period that is longer than δ_D (discussion of Lemma 7). Therefore, the defender moving with period δ_D and the attacker moving immediately is a unique equilibrium.

[4]Note that in the special case of $C_A = \mathcal{A}(\delta_D)$, the attacker has two best responses (moving immediately and never moving). Clearly, both players moving is an equilibrium as both play their best responses. On the other, if the attacker chooses not to move, then the defender's best response will either be a longer period δ'_D (the solution to $C_D = \mathcal{D}(\delta)$) or never moving. However, it is easy to see that $C_A < \mathcal{A}(\delta'_D)$, as $\mathcal{A}(l)$ is strictly increasing on (δ_D, ∞) since $\mathcal{A}(\delta_D) > 0$. Consequently, the attacker's best response would no longer be not to move. Therefore, both players moving is a unique equilibrium.

- On the other hand, if $C_A > \mathcal{A}(\delta_D)$, then the attacker's best response to the defender moving with period δ_D is to never move (Lemma 8). Thus, both players moving cannot be an equilibrium: if the attacker moves, then the defender's best response is to move with period δ_D, but the attacker's best response to this strategy is to never move. This leaves us with one possible equilibrium, in which the defender moves periodically and the targeting attacker never moves.

  Now, if $C_D = \mathcal{D}^N(l)$ has a solution δ'_D, then the defender's best response to the attacker not moving is to move periodically with period δ'_D. Furthermore, if $C_A \geq \mathcal{A}(\delta'_D)$, the attacker's best response to this period is not to move.[5] Thus, the defender moving with period δ'_D and the attacker not moving is an equilibrium.

- Finally, if $C_D = \mathcal{D}^N(l)$ does not have a solution, then the defender's best response to the targeting attacker never attacking is not to move (Lemma 7). On the other hand, if $C_D = \mathcal{D}^N(l)$ has a solution δ'_D and $C_A < \mathcal{A}(\delta'_D)$, then the attacker's best response is to attack immediately (Lemma 8). Consequently, the defender moving periodically and the targeting attacker moving immediately cannot be an equilibrium. Therefore, in this case, the game does not have an equilibrium.

□

4.3.3 Sequential Game: Deterrence by Committing to a Strategy

So far, I have modeled the mitigation of covert compromises as a simultaneous game. This is realistic for scenarios where neither the defender nor the targeting attacker can learn her opponent's strategy choice in advance. However, in practice, the defender can easily let the targeting attacker know about the defender's strategy by publicly announcing it. Even though one of the key elements of security is confidentiality, the defender can actually gain from revealing her strategy – as I will show in Section 4.4 – since this allows her to deter the targeting attacker from moving.

In this section, I model the conflict as a sequential game, where the defender chooses her strategy before the targeting attacker does. I assume that the defender announces her strategy (e.g., publicly commits herself to a certain cryptographic-key update policy) and the targeting attacker chooses her best response based on this knowledge. Furthermore, in this subsection, I restrict the defender's strategy set to the union of periodic strategies and not moving. The following theorem describes the defender's subgame-perfect equilibrium strategies.

Theorem 19. *Let δ_1 be the solution of $C_D = \mathcal{D}^A(\delta)$ (if it exists), δ_2 be the maximal period δ for which $C_A = \mathcal{A}(\delta)$, and δ_3 be the solution of $C_D = \mathcal{D}^N(\delta)$ (if it exists). In a subgame-perfect equilibrium, the defender's strategy is one of the following:*

- *not moving,*
- *periodic strategies with periods $\{\delta_1, \delta_2, \delta_3\}$.*

Based on the above theorem, one can easily find all subgame-perfect equilibria by iterating over the above strategies and, for each strategy, computing the targeting attacker's best response using Lemma 8, and finally comparing the defender's payoffs to find her equilibrium strategy (or strategies). Note that, for each case of Theorem 18, the set of possible equilibrium strategies in Theorem 19 could be restricted further. For example, in Case 2. (b) i., the only subgame perfect equilibrium is the defender moving periodically with δ'_D and the targeting attacker never moving. I defer the remaining cases to future work.

Proof. For the sake of contradiction, suppose that there exists a subgame-perfect equilibrium where the defender uses another period δ'.

First, assume that $\delta' > \delta_2$. In this case, I show that either period δ_1 or not moving yields a strictly greater payoff to the defender than period δ', given that the targeting attacker will always play her best-response strategy. From Lemma 8 and the definition of δ_2, we have that – in this case – the targeting attacker's unique best-response strategy against period δ' is to always attack immediately. From Lemma 6, we have that the defender's unique payoff-maximizing strategy against the targeting

[5] Again, in the special case of $C_A = \mathcal{A}(\delta'_D)$, the attacker has two best responses (similarly to footnote 4): moving immediately and never moving. If the attacker chooses not to move, we obviously have an equilibrium. On the other hand, if the attacker choose to move, the defender switches to the shorter period δ_D. But we already have that the attacker's best response to this strategy is not to move. Thus, both players moving cannot be an equilibrium.

attacker moving immediately is not moving if δ_1 does not exists, and a periodic strategy with the period δ_1 if it does exist. Now, if δ_1 does not exist or if it is greater than δ_2, then these strategies yield a strictly greater payoff than moving with period δ' due to their strictly lower moving costs, while keeping the targeting attacker's best-response strategy the same. On the other hand, if $\delta_1 \leq \delta_2$, then moving with period δ_1 deters the attacker from attacking. It is easy to see that this yields a higher payoff to the defender than period δ', as period δ_1 is a better response against the attacker moving immediately than period δ', and deterring the attacker only further decreases the defender's loss. Thus, period δ' cannot be an equilibrium strategy, which contradicts the initial supposition.

Second, assume that $\delta' < \delta_2$. In this case, I show that either period δ_2 or period δ_3 yields a strictly greater payoff to the defender than period δ', given that the targeting attacker will always play her best-response strategy. From Lemma 8 and the definition of δ_2, we have that – in this case – the targeting attacker's unique best-response strategy against period δ' is never to move. From Lemma 7, we have that the defender's unique payoff-maximizing strategy for the attacker never moving is not to move if δ_3 does not exist, and a periodic strategy with period δ_3 if it does exist. Now, if δ_3 exists and it is less than or equal to δ_2, then moving with period δ_3 instead of period δ' strictly increases the defender's payoff, while keeping the targeting attacker's best-response strategy the same. On the other hand, if δ_3 does not exist or if it is greater than δ_2, then the derivate of the defender's payoff as a function of her period is positive on $(0, \delta_2)$ (see the proofs of Lemmas 6 and 7). Consequently, the defender's payoff is strictly greater for any period in (δ', δ_2) than for δ'. Thus, δ' cannot be an equilibrium strategy, which contradicts the initial supposition. Therefore, the claim of the theorem must hold. □

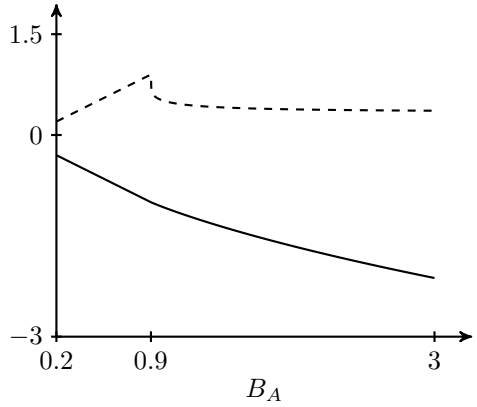
4.4 Numerical Illustrations

In this section, I present numerical results on the game. For the illustrations, I instantiate the model with the *exponential distribution* as the distribution F_A of the attack time. For rate parameter λ_A, the cumulative distribution function of the exponential distribution is $F_A(a) = 1 - e^{-\lambda_A a}$. For the remainder of this section, unless indicated otherwise, the parameters of the game are $C_D = C_A = B_A = \lambda_A = \lambda_N = 1$ and $B_N = 0.1$. Finally, I will refer to simultaneous-game Nash equilibria simply as equilibria, and I will refer to the defender's subgame-perfect equilibrium strategies as optimal strategies (because they maximize the defender's payoff given that the targeting attacker always plays her best response).

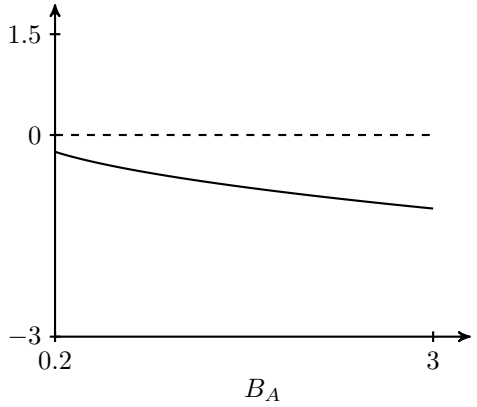
First, in Figure 4.3, I study the effects of varying how valuable the resource is, that is, varying the unit benefit B_A of the targeting attacker. Figure 4.3a shows the players' equilibrium payoffs as functions of B_A. The defender's equilibrium period for the same setup is shown by Figure 4.3c. We see in Figure 4.3a that the defender's payoff is strictly decreasing in B_A, which is unsurprising: the more valuable the resource is, the higher the potential losses are and the more investment in security is necessary. The targeting attacker's payoff, on the other hand, starts growing linearly, but then suffers a sharp drop, and finally converges to a finite positive value. This phenomenon is explained by the changes in the defender's equilibrium strategy at certain benefit values.

Figure 4.3c shows that, for lower values ($B_A < 0.9$), the defender does not protect the resource, as it is not valuable enough to be defended. Accordingly, the defender's period is not plotted in this range, and the targeting attacker's payoff is equal the value of the resource B_A. However, once the value reaches about 0.9, the defender starts protecting the resource. At this point, the attacker's payoff drops as she no longer has the resource compromised at all times. For higher values, the defender balances between losses due to being compromised and moving costs. Since the potential losses (i.e., how valuable the resource is) are increasing, the defender's cost rate is also increasing in B_A, which means that the time the resource is compromised decreases as its value increases. Hence, the targeting attacker's equilibrium payoff converges to zero.

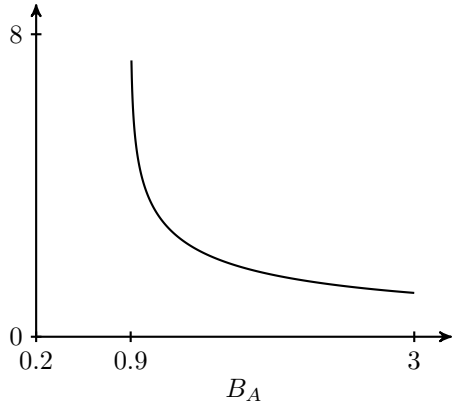
Figure 4.3b shows the players' payoffs for the defender's optimal strategy (and the targeting attacker's best response) as functions of B_A. The defender's optimal period for the same setup is shown by Figure 4.3d. This figure shows that the defender's strategy for this range of B_A is always to deter the targeting attacker from attacking. Hence, the targeting attacker's payoff is always zero in Figure 4.3b. To deter the targeting attacker, the defender is always using a period that is strictly shorter than her equilibrium period (i.e., she is moving faster than her equilibrium strategy), which can be seen by comparing Figures 4.3c and 4.3d. Finally, the key observation from comparing Figures 4.3a and 4.3b is that the defender's optimal payoff is much higher than her equilibrium payoff.



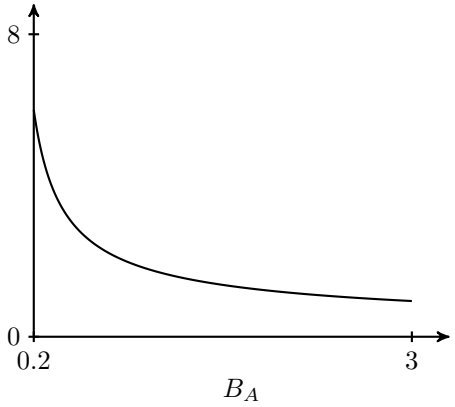
(a) The defender's and the targeting attacker's payoffs (solid and dashed lines, respectively) in equilibria as functions of B_A.



(b) The defender's and the targeting attacker's payoffs for the defender's optimal strategy as functions of B_A.



(c) The defender's equilibrium period as a function of B_A.



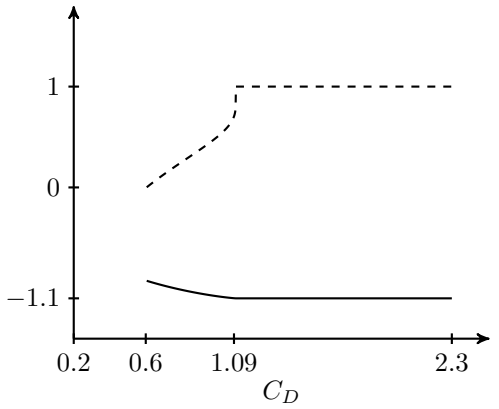
(d) The defender's optimal period as a function of B_A.

Figure 4.3: The effects of varying the unit benefit B_A of the targeting attacker.

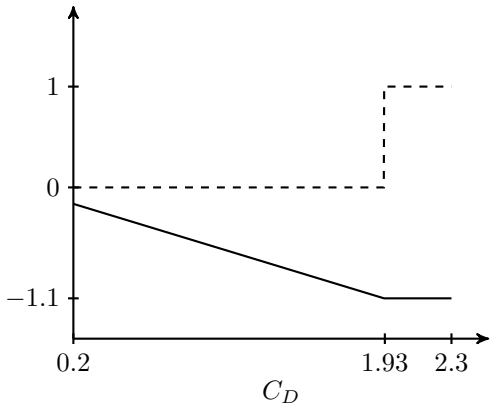
Second, in Figure 4.4, I study the effects of varying the defender's move cost C_D. Figure 4.4a shows the players' equilibrium payoffs as functions of C_D. The defender's equilibrium period for the same setup is shown by Figure 4.4c. We can see in Figure 4.4a that the defender's payoff is decreasing in C_D, while the targeting attacker's payoff is increasing. This is again unsurprising: the more costly defending the resource is, the greater the attacker's advantage is.

For lower costs ($C_D < 0.6$), the defender is at an overwhelming advantage, but there is no equilibrium (see Case 2. (b) ii. of Theorem 18). Accordingly, no values are plotted in this range. For costs between 0.6 and 1.09, no player is at an overwhelming advantage; hence, both players move from time to time. In this range, as the cost C_D increases, the defender's payoff decreases, while the targeting attacker's payoff increases. For higher costs ($C_D > 1.09$), the targeting attacker is at an overwhelming advantage. In this case, the defender never moves (i.e., gives up), while the attacker moves once at the beginning of the game. Hence, their payoffs are $B_A + B_N = -1.1$ and $B_A = 1$, respectively.

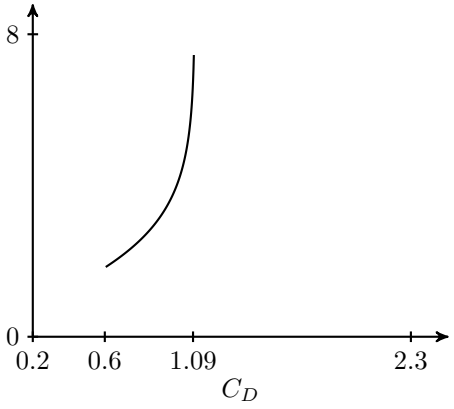
Figure 4.4b shows the players' payoffs for the defender's optimal strategy (and the targeting attacker's best response) as functions of C_D. The defender's optimal period is shown by Figure 4.4d. The figure shows that the defender's optimal strategy for move costs lower than 1.93 is to deter the targeting attacker. Hence, the targeting attacker's payoff is zero in this range. Since the unit benefit of the targeting attacker B_A is constant, the same constant period is used by the defender to deter the attacker, regardless of the move cost C_D. Consequently, as the move cost C_D increases, the cost of deterrence increases linearly and the defender's payoff decreases linearly. Finally, we see again that the defender's optimal payoff is much higher than her equilibrium payoff. However, for higher move costs ($C_D > 1.93$),



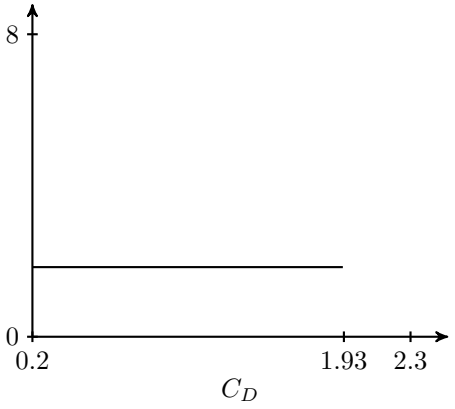
(a) The defender's and the targeting attacker's payoffs (solid and dashed lines, respectively) in equilibria as functions of C_D.



(b) The defender's and the targeting attacker's payoffs for the defender's optimal strategy as functions of C_D.



(c) The defender's equilibrium period as a function of C_D.



(d) The defender's optimal period as a function of C_D.

Figure 4.4: The effects of varying the defender's move cost C_D.

she must give up defending the resource, as in her equilibrium strategy for that range.

4.5 Related Work

4.5.1 Games of Timing

Previous cybersecurity-economics research has primarily focused on the choice of canonical actions to prevent, deter or otherwise mitigate incidents (e.g., [Laszka et al., 2012a]). However, being successful in dynamic environments shifts the focus from selecting the most suitable option from a pool of alternatives to a decision problem of *when* to act to get an advantage over an opponent. For example, in tactical security scenarios it is important to jump to action at the right time to avoid a loss of money or even human life (see, for example, timing of interventions in international conflicts). To understand these scenarios, so-called games of timing have been studied with the tools of non-cooperative game theory since the cold war era (see, for example, [Radzik and Orlowski, 1982, Zhadan, 1976]). For a detailed survey and summary of the theoretical contributions in this area, I refer the interested reader to [Radzik, 1996].

4.5.2 FlipIt: Modeling Targeted Attacks

Closely related to this study is the FlipIt model, which identifies optimal timing-related security choices under targeted attacks [van Dijk et al., 2013]. In FlipIt, two players compete for a resource that generates a payoff to the current owner. Players can make costly moves (i.e., "flips") to take ownership of the resource, however, they have to make moves under incomplete information about the current state of possession.

In the original FlipIt paper, equilibria and dominant strategies for simple cases of interaction are studied [van Dijk et al., 2013]. Other groups of researchers have worked on extensions [Pham and Cid, 2012, Laszka et al., 2013b]. For example, in [Laszka et al., 2013b], the FlipIt game is extended to the case where there are multiple resources. In addition, the usefulness of the FlipIt game has been investigated for various application scenarios [Bowers et al., 2012, van Dijk et al., 2013]. I detail the differences between the proposed model and FlipIt in Section 4.2.4.

FlipIt has also been studied in experiments where human participants were matched with computerized opponents [Nochenson and Grossklags, 2013]. This work has also been extended to consider different interface feedback modalities [Reitter et al., 2013]. The results complement the theoretical work by providing evidence for the difficulty to identify optimal choices when timing is the critical decision dimension.

4.6 Conclusions

In this chapter, I studied mitigation strategies for covert compromises of computing resources. Using game theory, I modeled a regime in which a defender must vie for a contested resource against both targeted and non-targeted covert attacks. As the main result, I found that periodic mitigation is the most effective strategy against both types of attacks and their combinations. Considering the simplicity of the periodic strategy, this result can be surprising, but it also serves as a theoretical justification for the prevalent periodic password and cryptographic-key renewal practices. Moreover, this result contradicts the lesson learned from the FlipIt model [van Dijk et al., 2013], which suggests that a defender facing an adaptive attacker should use an unpredictable, randomized strategy.

Furthermore, I also found that a defender is more willing to commit resources to defensive moves when being threatened by non-targeted and targeted attacks at the same time. This stands in contrast to the result that a high level of either threat type can force the defender to abandon defensive activities altogether.

Finally, I observed that there is a substantial difference between the defender's simultaneous and sequential (i.e., optimal) equilibrium strategies, both in the lengths of the equilibrium periods and the resulting payoffs. Therefore, a defender should not try to keep her strategy secret, but rather publicly commit to it.

4.7 Related Publications

- Laszka, A., Johnson, B., and Grossklags, J. (2013d). Mitigation of targeted and non-targeted covert attacks as a timing game. In *Proceedings of the 4th Conference on Decision and Game Theory for Security (GameSec)*, pages 175–191
- Laszka, A., Johnson, B., and Grossklags, J. (2013c). Mitigating covert compromises: A game-theoretic model of targeted and non-targeted covert attacks. In *Proceedings of the 9th Conference on Web and Internet Economics (WINE)*, pages 319–332

Chapter 5

Secure Team Composition

5.1 Introduction

In order to remain competitive, organizations and companies have to protect their trade secrets and business plans from unauthorized individuals. In information security, selectively restricting individuals' access to information is achieved using access control features and techniques. Providing effective access control in organizations has been referred to as the "traditional center of gravity of computer security" since it is a melting pot for human factors, systems engineering, and formal computer science approaches [Anderson, 2008]. Over the last decades, a large number of important contributions have been made to address various technical challenges to the problem of access control for important systems and sensitive data [Saltzer and Schroeder, 1975, Sandhu and Samarati, 1994]. This body of research is motivated in equal parts by the threat of malicious attackers from the outside and potential abuse by legitimate system users [Sindre and Opdahl, 2005]. One can further distinguish between those situations in which insiders exploit technical vulnerabilities of a system in opportunistic ways, and other situations in which employees abuse the trust placed in them [Anderson, 2008, Bishop, 2005]. In this chapter, I address the latter dimension of the problem space.

Data theft by trusted employees covers a significant share of insider attacks. For example, a CERT investigation of 23 attacks showed that "in 78% of the incidents, the insiders were authorized users with active computer accounts at the time of the incident. In 43% of the cases, the insider used his or her own username and password to carry out the incident" [Randazzo et al., 2005].

These attacks are occasionally attributed to disgruntled employees and are said to be primarily destructive in nature. However, the steady rise of cyber-espionage activities strongly motivates the threat scenario of employees stealing information for monetary rewards. A recent article summarized publicly-known United States legal data from the past four years and stated that "nearly 100 individual or corporate defendants have been charged by the Justice Department with stealing trade secrets or classified information" [Finn, 2013]. The article just considered theft benefiting one particular foreign nation. Therefore, it is reasonable to assume that the data merely represents the tip of the proverbial iceberg.

Turning a trusted employee into a spy provides a number of benefits for an outside attacker. First, a security compromise by an insider might not be discoverable in comparison to external network-based attacks that might leave traces identifiable for expert forensics teams. The result is that a corporation cannot adequately plan and respond to evidence of a stolen trade secret. Second, an insider can point the attacker towards particularly valuable secrets by identifying the so-to-speak needle in the haystack. Given the accelerating data growth within corporations it makes sense to assume that attackers are also suffering from information overload as a result of their successful but unguided network penetrations. Third, an insider can help the attacker interpret the stolen data through complementary communications that do not have to take place at the work location. Lastly, having an insider conduct the attack might be the only feasibly way for an attacker to circumvent the defenses of particularly well-defended targets such as military and intelligence services, i.e., the attacker makes use of the human as the weakest link.

In this chapter, I introduce a two-player non-deterministic game for modeling secure team selection

to add resilience against insider threats. A project manager, Alice, has a secret she wants to protect but must share with a team of individuals selected from within her organization; while an adversary, Eve, wants to learn this secret by bribing one potential team member. Eve does not know which individuals will be chosen by Alice, but both players have information about the bribeability of each potential team member. Specifically, the amount required to successfully bribe each such individual is given by a random variable with a known distribution but an unknown realization.

I describe the best-response strategies for both players and give necessary conditions for determining the game's equilibria. I find that Alice's best strategy involves minimizing the information available to Eve about the team composition. In particular, she should select each potential team member with a non-zero probability, unless she has a perfectly secure strategy. In the special case where the bribeability of each employee is given by a uniformly-distributed random variable, the equilibria can be divided into two outcomes – either Alice is perfectly secure, or her protection is based only on the randomness of her selection.

The remainder of this chapter is organized as follows. First, in Section 5.2, I introduce a game-theoretic model for team composition. Then, in Section 5.3, I provide general analytical results on the best-response and equilibrium strategies of the game. In Section 5.4, I instantiate the model with explicit distributions and study this special case. Finally, in Section 5.5, I discuss related work, and in Section 5.6 conclude the chapter.

5.2 Game-Theoretic Model

In this section, I describe a two-player, non-zero-sum, non-deterministic game which models the team composition scenario. First, I describe the general context and environment of the game and introduce the two players. Then, I define these players' pure strategy sets and their payoffs resulting from pure-strategy choices. Finally, I introduce notations to represent the players' mixed strategy spaces and express their expected payoffs in terms of these notations. For a simple illustration of the game's setup, see Figure 5.1.

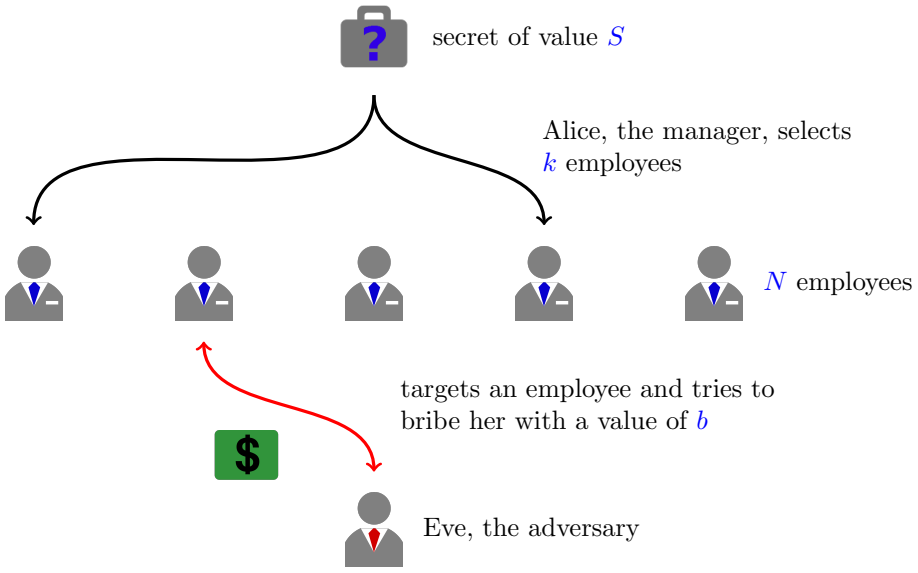


Figure 5.1: Illustration for the model with $N = 5$ and $k = 2$.

5.2.1 Environment

Suppose that an organization (or a company) with a secret of high value has N employees who are qualified to work on projects that require knowledge of the secret. The organization must share the secret with at least k employees in order to operate. The employees have varying levels of trustworthiness, which can only be estimated by the organization. For a given employee i, this uncertain trustworthiness level

is modeled as a random variable T_i, whose distribution $\mathcal{T}_i$ is known to all players. I explicitly disregard other constraints on team composition and assume that all aspects of the trustworthiness of an employee are captured by the random variable T_i. If $T_i = t_i$, then employee i will reveal what she knows whenever she is bribed with an amount greater than or equal to t_i, but she will never reveal the secret if she is bribed with an amount less than t_i. I use the standard cumulative distribution function notation

$$F_{T_i}(b) = \Pr[T_i \leq b] \tag{5.1}$$

to denote the probability that the trustworthiness level of employee i is at most b.

5.2.2 Players

The players in the game are called Alice and Eve. Alice is the project manager of an organization, who is responsible for selecting a team of qualified employees to work on a confidential project, which requires each team member to know a secret of the organization. This secret has a value of S, which is known to both players. Alice needs to share the secret with k of her N qualified employees to ensure the operation of the company.

Eve is a spy from either inside or outside of the organization. Eve wants to learn the secret and has the resources to bribe or eavesdrop on one of Alice's employees. If she eavesdrops on an employee, the trustworthiness level of the employee can be interpreted as a measure of the difficulty of eavesdropping on that employee. Note that Eve does not know which employees are on the team.

5.2.3 Pure-Strategy Sets

Alice's pure-strategy choice is to select exactly k of her N employees with whom she shares the secret. Formally, she chooses a size-k subset I of $\{1, \ldots, N\}$.

Eve's pure-strategy choice is to select one employee and an amount to bribe with. Formally, she chooses a pair (i, b) consisting of an employee index $i \in \{1, \ldots, N\}$ and a bribe value $b \in \mathbb{R}_{\geq 0}$. Note that the players do not know which pure strategy their opponent has selected.

5.2.4 Payoffs

Suppose that Alice plays a pure strategy I, and Eve plays a pure strategy (i, b). If $i \in I$ and $T_i \leq b$, then Eve wins the value of the secret minus the amount of the bribe, and Alice loses the value of the secret. In all other cases, Eve loses the amount of the bribe, and Alice loses nothing.[1] The payoffs for the different scenarios are summarized by Table 5.1.

Table 5.1: Payoffs for a Pure-Strategy Profile $I, (i, b)$

Strategy profile and outcome	Payoff for Alice	Payoff for Eve
$i \in I$ and $T_i \leq b$	$-S$	$S - b$
$i \notin I$ or $T_i > b$	0	$-b$

5.2.5 Representation of Mixed Strategies

A player's mixed strategy is a distribution over the set of her pure strategies. For Alice, the canonical representation of her mixed-strategy space is a finite probability distribution over the set of size-k subsets of $\{1, \ldots, N\}$. For Eve, the canonical representation of her mixed strategy space is a continuous probability distribution over the set $\{1, \ldots, N\} \times \mathbb{R}_{\geq 0}$. Because of the structure of the game, the expected payoffs for both players can be determined by representations of the mixed-strategy spaces that are simpler than the canonical ones. In the following subsections, I introduce these representations and use them to express the players' expected payoffs.

[1] Note that I assume that the secret is equally valuable to both players only for notational simplicity. The best response strategies and – consequently – the equilibrium profiles remain the same if we apply an element-wise positive-affine transformation to the players' payoffs.

Alice's Mixed Strategies

In the canonical representation of Alice's mixed strategy, we would let α_I denote the probability that she recruits the members of the size-k set I into the project team. However, since Eve can bribe only one employee, the payoffs for any mixed-strategy profile depend only on the probabilities of Alice sharing the secret with each employee. More specifically, the probability of Eve learning the secret is

$$\begin{aligned}\sum_{i\in\{1,\dots,N\}} &\Pr[\text{Alice shares the secret with } i]\cdot\\ &\Pr[\text{Eve succeeds with her bribe} \mid \text{Eve targets } i]\cdot\\ &\Pr[\text{Eve targets } i]\ .\end{aligned} \tag{5.2}$$

In other words, Alice's strategy choice influences the payoffs only through the probabilities of sharing with each employee, not the actual distribution over the subsets. Since several different mixed strategies might induce the same projection onto employee probabilities, we can gain simplicity by restricting our attention to these projections. The goal here is to describe the space of these projections.

For each $i = 1, \dots, N$, let a_i denote the probability that Alice shares the secret with employee i. Formally, let

$$a_i = \sum_{I:\ i\in I} \alpha_I\ . \tag{5.3}$$

The requirement that Alice has to share the secret with k employees induces the notational constraint

$$\sum_{i=1}^{N} a_i = k\ . \tag{5.4}$$

It is easy to see that, for any mixed strategy of Alice, the projection vector $\boldsymbol{a}$ satisfies $0 \le a_i \le 1$ for every i, and $\sum_{i=1}^{N} a_i = k$. However, it remains to show that this is also true vice versa, that is, to show that, for any vector $\boldsymbol{a}$ of N probabilities whose sum is k, there exists a mixed strategy for Alice whose projection is $\boldsymbol{a}$. The following theorem shows that this is indeed true.

Theorem 20. *For any vector of probabilities $\boldsymbol{a}$ that satisfies $\sum_i a_i = k$, there exists a mixed strategy $\boldsymbol{\alpha}$ for Alice such that, for every i, the probability of sharing the secret with employee i is a_i. Furthermore, there is such a mixed strategy whose support consists of at most N sets.*

Proof. The proof is constructive, and it is based on the following algorithm.

1. For every k-subset I, let $\alpha_I = 0$.
2. Let I be a k-subset consisting of the positions with the k highest a_i (if there are multiple such subsets, select an arbitrary one).
3. Let p be the maximum value subject to
   - for every $i \in I$, $a_i - p \ge 0$ and
   - for every $i \notin I$, a_i satisfies the MaxProb constraint (for the definition of this constraint, see below).
4. Increase α_I by p and, for every $i \in I$, decrease a_i by p.
5. If there is an $a_i > 0$, then continue from Step 2.

Now, I introduce the MaxProb constraint. First, notice that a non-negative vector $\boldsymbol{a}$ has to satisfy two necessary constraints to be a mixed strategy over k-subsets: $\sum_i a_i = k$ and, for every i, $a_i \le 1$. It is easy to see that a vector cannot be a mixed strategy over k-subsets if it violates one of the constraints. Similarly, at any step of the algorithm's execution, it has to hold that $a_i \le k'$ for every i, where $k' = \sum_i a_i/k$. From this, we can formulate the MaxProb constraint as $p \le \sum_j a_j/k - a_i$. Finally, I call a vector $\boldsymbol{a}$ proper if, for every i, $a_i \ge 0$ and $a_i \le k'$. Obviously, we have that the input vector is proper.

Next, I prove the correctness of the algorithm. First, it is easy to see that the vector $\boldsymbol{a}$ stays non-negative (first constraint of Step 3). Second, we can show that the vector $\boldsymbol{a}$ stays proper. Every element $i \in I$ is decreased by p, but the sum is decreased by $k \cdot p$; thus, if the elements of I satisfied $a_i \le \sum_j a_j/k$

before the decrease, they still satisfy it after the decrease. As for the non-elements $i \notin I$, the MaxProb constraint ensures that the vector stays proper. Third, it is easy to see that if a vector is proper and non-zero, then it has at least k positive elements (as no element can be higher than the sum over k). Fourth, it can be shown that if there are k positive elements, then the maximum p of Step 3 has to be positive (as there are at most k elements for which the equality $a_i = \sum_j a_j/k$ holds; hence, $p = \sum_j a_j/k - a_i$ does not hold for $p = 0$ and $i \notin I$).

Note that, at this point, we already have that the algorithm starts with a proper non-zero vector, it decreases the elements (possibly an infinite number of times) keeping the vector proper and non-negative, and finally decreases the last k positive elements to zero at once. It remains to show that the algorithm terminates after a finite number of iterations. However, we can do much better than that. Let M be the set of elements i for which the equality $a_i = \sum_j a_j/k$ holds (i.e., the set of maximal elements), let Z be the set of zero elements, and let O be the set of elements neither in M nor in Z. First, if an element belongs to Z, then it obviously remains there after a decrease. Second, if an element belongs to M, then it remains there after a decrease (as any element of M has to be a member of I). Third, in every iteration, at least one element of O is moved to either M or Z (as one of the constraints of Step 3 has to be an equality for at least one element for the maximum p). Fourth, $|O| \leq N$ trivially. Therefore, there are at most N iterations, as we remove an element from the set O in every iteration and $|O|$ is at most N initially. Notice that this also implies that the cardinality of the resulting distribution's support (the number of k-subsets with non-zero probability) is also at most N.

Finally, we have to show that the resulting $\boldsymbol{\alpha}$ is indeed a distribution, but this is very easy. First, $\sum_I \alpha_I = 1$, as $\sum_i a_i = k$ initially and we decrease it by $k \cdot p$ when we assign p probability to one of the subsets. Second, for every i, $\sum_{I \ni i} \alpha_I = a_i$, as we increase the probability of a containing subset by p when we decrease the value of a_i by p. □

Note that I not only proved the existence of a mixed strategy, but also devised an algorithm for finding a simple one. This is important from a practical point of view, as I will establish results in Section 5.3 on best-response and equilibrium strategies based on the projection representation. The above algorithm can be used in practice to find a feasible mixed strategy. Finally, note that there can exist multiple mixed strategies with the same projection representation.[2] Hence, there does not exist a unique mixed strategy for every projection, which is unsurprising considering that the projections were meant to be a more compact representation.

Eve's Mixed Strategies

To represent Eve's mixed strategies, which are distributions over the set $\{1, \ldots, N\} \times \mathbb{R}_{\geq 0}$, I introduce two random variables, Y and B. Random variable Y takes values in $\{1, \ldots, N\}$, and it represents which employee Eve has chosen to bribe. Random variable B takes values in $\mathbb{R}_{\geq 0}$, and represents the amount of the bribe.

Similarly to the representation of Alice's mixed strategies, for each $i = 1, \ldots, N$, let e_i to be the probability that Eve bribes employee i, so that we have

$$e_i = \Pr[Y = i] \ . \tag{5.5}$$

Since Eve always chooses exactly one employee, we have

$$\sum_{i=1}^{N} e_i = 1 \ . \tag{5.6}$$

Using the standard cumulative distribution function notation, a distribution over bribe values can be described as

$$F_B(b) = \Pr[B \leq b] \ , \tag{5.7}$$

which gives the probability that the value of the bribe chosen by Eve is at most b. It is also useful to describe the conditional distributions over bribes focused on a particular employee i. For each $i =$

[2] For an example, consider $N = 4$ and $k = 2$. Selecting the first and the second employees with %50 probability and the third and the fourth with %50 has the same projection as selecting the first and the third employees with %50 probability and the second and the fourth with %50.

$1, \ldots, N$, let B_i be the random variable whose range is the set of all possible bribes to player i, and whose distribution $\mathcal{B}_i$ is defined by

$$F_{B_i}(b) = \Pr[B_i \leq b] = \Pr[B \leq b | Y = i] \ . \tag{5.8}$$

In what follows, Eve's mixed strategies will be represented as pairs $(\boldsymbol{e}, \boldsymbol{\mathcal{B}})$, where each e_i is the probability that Eve bribes employee i, and each $\mathcal{B}_i$ is a distribution over bribe values, conditioned on the assumption that Eve chooses to bribe employee i.

5.2.6 Expected Payoffs for Mixed Strategies

In order to use the simplified mixed-strategy representation defined above, we have to express the players' expected payoffs in terms of these representations. If Alice plays a mixed strategy represented by $\boldsymbol{a}$ and Eve plays a mixed strategy represented by $(\boldsymbol{e}, \boldsymbol{\mathcal{B}})$, then the expected payoff for Alice is

$$-S \cdot \sum_{i=1}^{N} a_i \cdot e_i \cdot \Pr[T_i \leq B_i] \ , \tag{5.9}$$

and the expected payoff for Eve is

$$S \cdot \sum_{i=1}^{N} (a_i \cdot e_i \cdot \Pr[T_i \leq B_i]) - \sum_{i=1}^{N} e_i \cdot E[B_i] \ , \tag{5.10}$$

where $E[B_i]$ denotes the expected value of B_i under the distribution $\mathcal{B}_i$.

5.3 Analytical Results

In this section, I derive analytical results on the structure of the game's Nash equilibria. I begin with describing Alice's and Eve's best-response strategies. Then, I use these results to constrain the players' strategies in an equilibrium. Finally, based on these constraints, I provide results on the existence and uniqueness of the equilibrium strategies and payoffs.

5.3.1 Best-Response Strategies

Alice's Best Response

For a fixed strategy of Eve, Alice's best response minimizes the probability of the secret being compromised. Since the probability of employee i being targeted and successfully bribed is $e_i \cdot \Pr[T_i < B_i]$, Alice has to choose a set I of k employees that minimizes $\sum_{i \in I} e_i \cdot \Pr[T_i \leq B_i]$. However, as the set of k employees minimizing the probability of the secret being disclosed may be non-unique, Alice's best response can be a mixed strategy $\boldsymbol{a}$ whose support consists of more than k employees. This notion is formalized by the following lemma.

Lemma 9. *Given Eve's mixed strategy $(\boldsymbol{e}, \boldsymbol{\mathcal{B}})$, any best-response strategy $\boldsymbol{a}$ for Alice has to satisfy the following properties.*

- *For any employee i, if there are at least $N - k$ employees whose probabilities of being targeted and successfully bribed are strictly greater than that of i, then $a_i = 1$.*
- *For any employee i, if there are at least k employees whose probabilities of being targeted and successfully bribed are strictly less than that of i, then $a_i = 0$.*

Proof. First, for any employee i, if there are at least $N - k$ employees whose probabilities of sharing the secret are strictly greater than that of i, then i is a member of every size-k subset of employees that minimizes the probability of the secret being disclosed. Thus, in any best response, Alice always shares the secret with this employee i.

Second, for any employee i, if there are at least k employees whose probabilities of sharing he secret are strictly less than that of i, then i cannot be a member of any k-subset that minimizes the probability of the secret being disclosed. Thus, i cannot be in the support of any mixed strategy that is a best response for Alice. □

Eve's Best Response

Suppose that Alice is playing a mixed strategy where a_i is the probability that she shares the secret with employee i. Let $\text{MaxUE}(\mathcal{T}_i, a_i)$ denote the maximum payoff that Eve can attain from targeting employee i. Formally, let

$$\text{MaxUE}(\mathcal{T}_i, a_i) = \max_{b \in \mathbb{R}_{\geq 0}} \left(a_i \cdot S \cdot \Pr[T_i \leq b] - b\right) \ . \tag{5.11}$$

Lemma 10. *For any employee i and trustworthiness distribution $\mathcal{T}_i$, Eve's maximum payoff* $\text{MaxUE}(\mathcal{T}_i, a_i)$ *as a function of Alice's secret-sharing probability a_i has the following properties:*

1. $\text{MaxUE}(\mathcal{T}_i, 0) = 0$,
2. $\text{MaxUE}(\mathcal{T}_i, x)$ *is increasing in x,*
3. *if* $\text{MaxUE}(\mathcal{T}_i, z) > 0$ *for some z, then* $\text{MaxUE}(\mathcal{T}_i, x)$ *is strictly increasing in x on $(z, 1]$,*
4. $\text{MaxUE}(\mathcal{T}_i, x)$ *is uniformly continuous in x.*

Proof.

1. First, it is clear that the maximum of $\text{MaxUE}(\mathcal{T}_i, 0) = 0 \cdot S \cdot \Pr[T_i \leq b] - b = -b$, given that $b \in \mathbb{R}_{\geq 0}$, is attained at $b = 0$.

2. To show that the function is increasing in x, let $x, y \in [0, 1]$ with $x < y$. Let b_x be a bribe value at which the maximum payoff is attained for secret-sharing probability x, that is, $\text{MaxUE}(\mathcal{T}_i, x) = x \cdot S \cdot \Pr[T_i \leq b_x] - b_x$. Then, we have

$$\begin{aligned} \text{MaxUE}(\mathcal{T}_i, y) &\geq y \cdot S \cdot \Pr[T_i \leq b_x] - b_x && (5.12)\\ &\geq x \cdot S \cdot \Pr[T_i \leq b_x] - b_x && (5.13)\\ &= \text{MaxUE}(\mathcal{T}_i, x) \ . && (5.14) \end{aligned}$$

3. To show that the function is strictly increasing in x on $(z, 1]$, let $x, y \in (z, 1]$ with $x < y$. Let b_x be a bribe value at which the maximum payoff is attained for secret-sharing probability x, that is, $\text{MaxUE}(\mathcal{T}_i, x) = x \cdot S \cdot \Pr[T_i \leq b_x] - b_x$. Since $\text{MaxUE}(\mathcal{T}_i, x) \geq \text{MaxUE}(\mathcal{T}_i, z) > 0$ (see previous case), we have $\Pr[T_i \leq b_x] > 0$. Then,

$$\begin{aligned} \text{MaxUE}(\mathcal{T}_i, y) &\geq y \cdot S \cdot \Pr[T_i \leq b_x] - b_x && (5.15)\\ &> x \cdot S \cdot \Pr[T_i \leq b_x] - b_x && (5.16)\\ &= \text{MaxUE}(\mathcal{T}_i, x) \ . && (5.17) \end{aligned}$$

4. Finally, to show uniform continuity in x, let $x, y \in [0, 1]$ with $x < y$, and let b_y be a bribe value at which the maximum payoff is attained for secret-sharing probability y, that is, $\text{MaxUE}(\mathcal{T}_i, y) = y \cdot S \cdot \Pr[T_i \leq b_y] - b_y$. Using the previous result that $\text{MaxUE}(\mathcal{T}_i, y)$ is increasing, we have

$$\begin{aligned} 0 &< \text{MaxUE}(\mathcal{T}_i, y) - \text{MaxUE}(\mathcal{T}_i, x) && (5.18)\\ &\leq (y \cdot S \cdot \Pr[T_i \leq b_y] - b_y) - (x \cdot S \cdot \Pr[T_i \leq b_y] - b_y) && (5.19)\\ &= (y - x) \cdot S \cdot \Pr[T_i \leq b_y] && (5.20)\\ &\leq (y - x) \cdot S \ . && (5.21) \end{aligned}$$

   So $\text{MaxUE}(\mathcal{T}_i, x)$ satisfies a Lipschitz condition in the variable x with Lipschitz constant S; and hence, it is uniformly continuous.

□

For a given employee, it is possible for more than one bribe value to give Eve the maximal payoff. Let $\text{ArgMaxBE}(\mathcal{T}_i, x)$ denote the set of bribes that give Eve her maximum payoff for employee i, which is a function of the employee's trustworthiness level distribution and the probability of receiving the secret from Alice. Formally,

$$\text{ArgMaxBE}(\mathcal{T}_i, a_i) = \operatorname*{argmax}_{b \in \mathbb{R}_{\geq 0}} \left(a_i \cdot S \cdot \Pr[T_i \leq b] - b\right) \ . \tag{5.22}$$

Using this notation, we may define constraints on Eve's best response strategy as follows.

Lemma 11. *Given Alice's mixed strategy $\boldsymbol{a}$, Eve's best response selects an employee i with the largest* $\mathrm{MaxUE}(\mathcal{T}_i, a_i)$ *over all $i \in \{1, \ldots, N\}$, and then chooses a bribe value b from* $\mathrm{ArgMaxBE}(\mathcal{T}_i, a_i)$*. If there are multiple pairs (i, b) satisfying these constraints, then Eve may choose any distribution whose support is a subset of these payoff-maximizing pure strategies.*

Proof. Follows readily from Equations (5.10), (5.11), and (5.22). □

5.3.2 Nash Equilibria

Above, I introduced constraints on the players' best-response strategies. In this subsection, I introduce additional constraints on their equilibrium strategies.

Alice's Strategy in an Equilibrium

It is generally in Alice's interest to minimize the maximum attainable payoff for Eve, as this generally (but, since the game is non-zero sum, not necessarily) minimizes her loss. We know that Eve's best response is always to choose an employee (or a set of employees) which will maximize $\mathrm{MaxUE}(\mathcal{T}_i, a_i)$ over i. Therefore, in an equilibrium, Alice's strategy should try to equalize these quantities, subject to the constraints that her sharing probabilities cannot exceed 1 and that they sum to k.

This notion is made formal in the following theorem.

Theorem 21. *In any Nash equilibrium, Alice's strategy satisfies the following constraints.*

1. *For any pair of employees i and j, if $a_i, a_j < 1$, then* $\mathrm{MaxUE}(\mathcal{T}_i, a_i) = \mathrm{MaxUE}(\mathcal{T}_j, a_j)$.
2. *For any pair of employees i and j, if $a_j < a_i = 1$, then* $\mathrm{MaxUE}(\mathcal{T}_i, a_i) \leq \mathrm{MaxUE}(\mathcal{T}_j, a_j)$.

Proof. Let $\boldsymbol{a}$ and $(\boldsymbol{e}, \boldsymbol{\mathcal{B}})$ be Alice's and Eve's mixed strategies and assume that this strategy profile is a Nash equilibrium.

1. For the sake of contradiction, suppose that $a_i, a_j < 1$ and it holds that $\mathrm{MaxUE}(\mathcal{T}_i, a_i) \neq \mathrm{MaxUE}(\mathcal{T}_j, a_j)$. We can assume without loss of generality that $\mathrm{MaxUE}(\mathcal{T}_i, a_i) < \mathrm{MaxUE}(\mathcal{T}_j, a_j)$. Then, $\mathrm{MaxUE}(\mathcal{T}_j, a_j) > 0$, which (from Lemma 10.1) implies that $a_j > 0$. From Lemma 11, we have that the support of Eve's best-response mixed strategy does not include i. Thus, Alice may strictly increase a_i towards 1, and strictly decrease every other non-zero component of her strategy for employees other than i, while still satisfying the constraint $\sum_m a_m = k$. By decreasing her secret-sharing probability on every employee that Eve might bribe, Alice necessarily decreases the total probability of Eve learning the secret. Therefore, Alice can improve her expected payoff by changing her strategy, which contradicts the equilibrium condition.

2. For the sake of contradiction, suppose that $a_j < a_i = 1$ and that $\mathrm{MaxUE}(\mathcal{T}_i, a_i) > \mathrm{MaxUE}(\mathcal{T}_j, a_j)$. Then, $\mathrm{MaxUE}(\mathcal{T}_i, a_i) > 0$, which (based on Lemma 10) implies that $a_i > 0$. Consequently, we have (from Lemma 11) that the support of Eve's mixed strategy does not include employee j. So Alice may simultaneously increase a_j towards 1 and decrease her non-zero secret-sharing probabilities for employees other than j, all while satisfying the constraint $\sum_m a_m = k$. Again, by decreasing her secret-sharing probability on every employee that Eve might bribe, Alice necessarily decreases the total probability of Eve learning the secret. Hence, this strategy change will increase her expected payoff, contradicting the equilibrium condition.

□

It follows from Theorem 21 that Alice's equilibrium strategy $\boldsymbol{a}$ may have some employees with whom she shares the secret with certainty, but for all other employees, her secret-sharing distribution is only constrained by a smoothness constraint on the quantities $\mathrm{MaxUE}(\mathcal{T}_i, a_i)$. Furthermore, these quantities do not depend on Eve's strategy, a fact on which we can rely when computing an equilibrium.

From Theorem 21, we also have the following readily.

Corollary 3. *In any Nash equilibrium,*

- *Alice is either perfectly secure, that is, Eve has no strategy against her with a positive payoff, or else Alice shares the secret with every employee with a non-zero probability. Formally, either* $\mathrm{MaxUE}(\mathcal{T}_i, a_i) = 0$ *for every employee* i*, or* $a_i > 0$ *for every employee* i*.*

- *The employees with whom Alice shares the secret with certainty are at most as likely to be targeted by Eve as the other employees, with whom Alice is less likely to share the secret.*

It is interesting to compare the first point of the above corollary with Lemma 11. The former says that Alice shares the secret with every employee with a non-zero probability (when she cannot be secure), while Lemma 11 says that Alice never shares the secret with an employee if there are at least k employees that have lower probabilities of being targeted and successfully bribed. Since an equilibrium strategy is necessarily a best response, it has to satisfy both constraints. This implies that, in an equilibrium, Eve equalizes the probability of targeting and successfully bribing over the set of employees that maximize her expected payoff.

Eve's Strategy in an Equilibrium

In this section, I build on the constraints on Alice's equilibrium strategies presented in Theorem 21 to describe Eve's equilibrium strategies. In the previous paragraph, I argued that, in an equilibrium, Eve equalizes the probability of targeting and successfully bribing over the set of employees that maximize her payoff.

This notion is made formal by the following theorem.

Theorem 22. *In a Nash equilibrium, if* $a_i, a_j < 1$ *for a pair of employees* i *and* j*, then* $e_i \cdot \Pr[T_i \leq B_i] = e_j \cdot \Pr[T_j \leq B_j]$.

Proof. Let $\boldsymbol{a}, (\boldsymbol{e}, \boldsymbol{\mathcal{B}})$ be Alice's and Eve's mixed strategies, and assume that this strategy profile is a Nash equilibrium. For the sake of contradiction, suppose that $e_i \cdot \Pr[T_i \leq B_i]$ is non-uniform over the set of employees with whom Alice does not always share the secret. Furthermore, let I_{max} be the set of employees i for which $e_i \cdot \Pr[T_i \leq B_i]$ is maximal.

First, suppose that $k \leq N - |I_{max}|$. Then, Alice's best response never shares the secret with the employees in I_{max}, that is, $a_i = 0$ for all $i \in I_{max}$, as there are k strictly better employees (as stated in Lemma 9). Consequently, we have $e_i = 0$ for every $i \in I_{max}$, as Eve's strategy also has to be a best response. But this implies that $e_i \cdot \Pr[T_i \leq B_i] = 0$ for every i such that $a_i < 1$, which contradicts that $e_i \cdot \Pr[T_i \leq B_i]$ is non-uniform. Thus, it has to hold that $k > N - |I_{max}|$.

From $k > N - |I_{max}|$, we have that Alice's best response always shares the secret with every employee i for which $e_i \cdot \Pr[T_i \leq B_i]$ is not maximal (as stated in Lemma 9). Consequently, the only employees i for which $a_i < 1$ holds are the employees in I_{max}. But this contradicts that $e_i \cdot \Pr[T_i \leq B_i]$ is non-uniform since all employees in I_{max} have the same maximal $e_i \cdot \Pr[T_i \leq B_i]$. □

5.3.3 Existence and Multiplicity of Equilibrium Strategies and Payoffs

In the previous subsections, I have formulated constraints on the equilibria of the game. Here, I provide existence and uniqueness results on the equilibrium strategies and payoffs.

I begin with showing that there always exists at least on equilibrium strategy profile.

Theorem 23. *The game always has at least one Nash equilibrium.*

Proof. The proof is constructive, that is, I show the existence of an equilibrium strategy profile by providing an algorithm for computing one. Based on Theorems 21 and 22, I devise the following algorithm.

1. *Find an equilibrium strategy* $\boldsymbol{a}^*$ *for Alice*:
   I begin with finding a mixed-strategy $\boldsymbol{a}^*$ that satisfies Theorem 21. Since we have from Lemma 10 that every $\mathrm{MaxUE}(\mathcal{T}_i, a_i)$ is increasing and uniformly continuous in a_i, there always exists a solution $\boldsymbol{a}^*$ satisfying the constraints of Theorem 21.[3]

[3] Note that, since $\mathrm{MaxUE}(\mathcal{T}_i, a_i)$ is not strictly increasing, the solution might not be unique. I deal with uniqueness in the subsequent theorem.

2. *Find an equilibrium strategy* $(\boldsymbol{e}^*, \boldsymbol{\mathcal{B}}^*)$ *for Eve*:
   I continue with finding a mixed-strategy $(\boldsymbol{e}^*, \boldsymbol{\mathcal{B}}^*)$ that satisfies both Lemma 11 and Theorem 22. Let $MaxUE^* = \max_i \mathrm{MaxUE}(\mathcal{T}_i, a_i^*)$ and let I^* be the set of employees for whom the maximum is attained. If $MaxUE^* = 0$, then there is no strategy with a positive expected payoff for Eve, so we let $B_i^* \equiv 0$ for every i (and $\boldsymbol{e}^*$ can be an arbitrary distribution). Otherwise, we find a strategy which gives Eve an expected payoff of $MaxUE^*$ and which ensures that Alice will not deviate from her strategy as follows.

   (a) For every $i \notin I^*$, let $e_i^* = 0$.

   (b) For every $i \in I^*$, choose an arbitrary bribe value from $\mathrm{ArgMaxBE}(\mathcal{T}_i, a_i^*)$ and let B_i^* always take this value. Finally, let

$$e_i^* = \frac{\frac{1}{\Pr[T_i \leq B_i^*]}}{\sum_j \frac{1}{\Pr[T_j \leq B_j^*]}} \,. \tag{5.23}$$

I now prove that the mixed-strategy profile $\boldsymbol{a}^*, (\boldsymbol{e}^*, \boldsymbol{\mathcal{B}}^*)$ forms an equilibrium, i.e., that both strategies are best responses to each other. First, if $MaxUE^* = 0$, then we have the claim readily, as Eve cannot achieve an expected payoff higher than 0. Now, assume that $MaxUE^* > 0$. First, it is easy to see that Eve's strategy is indeed a best-response, as she targets the employees which give her maximal expected payoff and bribes them with optimal bribe values.

Second, I show that Alice's strategy is a best-response. Observe that, in Step (2b), we have chosen a distribution for Eve so that the probability of targeting and successfully bribing is uniform over employees in I^* and 0 for employees not in I^*. Since $\boldsymbol{a}^*$ satisfies Theorem 21, we also have that Alice shares the secret with a probability less than one with employees in I^* and with a probability of one with employees not in I^*. As Alice's best response is to share the secret with those employees whose probabilities of being targeted and successfully bribed are the lowest, we have that $\boldsymbol{a}^*$ is indeed a best response. □

The algorithm presented above proves that an equilibrium always exists; however, it can also be used to compute an equilibrium in practice. The challenge in this case lies in finding strategies that satisfy the constraints given by Theorems 21 and 22, respectively. Fortunately, this can be performed, for example, using any multidimensional numerical optimization method (e.g., the Nelder-Mead algorithm [Nelder and Mead, 1965] or some modern optimization method) by using the sum of the amounts by which each constraining equality is violated as the objective function. Note that the actual time complexity of finding an equilibrium depends on the trustworthiness level distributions.[4]

The next theorem shows that Alice's equilibrium strategy is essentially unique, which implies the uniqueness of Eve's equilibrium payoff.

Theorem 24. *If Alice has no perfectly secure strategy, then the projection representation* $\boldsymbol{a}$ *of her equilibrium strategies is unique.*

Proof. For the sake of contradiction, suppose that the claim of the theorem does not hold, that is, there exist two distinct equilibrium strategies $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$. From Theorem 21, we have that Eve's maximum payoff MaxUE for targeting an employee is uniform over the employees with whom Alice does not certainly share the secret. Let these uniform maximum payoffs for the strategies $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$ be u^1 and u^2, respectively. Note that, since Alice has no perfectly secure strategy, we have $u^1 > 0$ and $u^2 > 0$.

First, suppose that $u^1 = u^2$. Since $\mathrm{MaxUE}(\mathcal{T}_i, a_i)$ is strictly increasing, there exists only one a_i for each i such that $\mathrm{MaxUE}(\mathcal{T}_i, a_i) = u^1$. Thus, we have $a_i^1 = a_i^2$ for the employees who have a sharing probability lower than 1. On the other hand, the set of employees who have a sharing probability of 1 has to be equal for the two strategies, since an employee i for whom $\mathrm{MaxUE}(\mathcal{T}_i, a_i) < u^1 = u^2$ is always in this set. Thus, we have $a_i^1 = a_i^2$ for every employee. However, this leads to a contradiction with the initial supposition that $\boldsymbol{a}^1 \neq \boldsymbol{a}^2$.

Second, suppose that $u^1 > u^2$. For every employee i with $a_i^2 = 1$, we have $a_i^1 = 1$, as $\mathrm{MaxUE}(\mathcal{T}_i, a_i^1) < u^1 < u^2$. Thus, we have $a_i^1 = a_i^2$ for these employees. On the other hand, for every employee i with $a_i^2 < 1$ (i.e., $\mathrm{MaxUE}(\mathcal{T}_i, a_i^2) = u^2$), we either have $a_i^1 = 1$ or $\mathrm{MaxUE}(\mathcal{T}_i, a_i^1) = u^1$. In the second case,

[4] For instance, if we choose a cumulative function whose values are hard to compute, then the problem of computing an equilibrium is obviously hard as well. However, since the distributions are more likely to be based on industry-wide beliefs, empirical measures, and estimates based on statistical data, etc., than pathological functions, this is not a practical problem. Furthermore, since the distributions themselves are estimates, it actually suffices to only estimate an equilibrium.

we have $a_i^1 > a_i^2$ since $\text{MaxUE}(\mathcal{T}_i, a_i)$ is strictly increasing. Thus, we have $a_i^1 > a_i^2$ for these employees. However, this leads to the contradiction $1 = \sum_i a^1 > \sum_i a^2 = 1$.

Finally, the case $u^1 < u^2$ leads to a contradiction for the same reasons as the previous case. Therefore, the claim of the theorem has to hold. □

Finally, based on the above theorem, I show the uniqueness of Eve's equilibrium payoff.

Corollary 4. *Eve's equilibrium payoff is always unique.*

Proof. If Alice has only perfectly secure strategies, then Eve's equilibrium payoff is 0. On the other hand, if Alice has no perfectly secure strategy, then we have from Theorem 24 that her strategy and, hence, Eve's payoff has to be unique. Thus, it remains to show that non-secure and secure equilibrium strategies for Alice cannot exist at the same time.

For the sake of contradiction, suppose that this is not true, that is, there exist a non-secure and a secure strategy, denoted by $\boldsymbol{a}^1$ and $\boldsymbol{a}^2$. First, for every employee with $a_i^1 = 1$, we obviously have $a_i^1 \geq a_i^2$. Second, for every employee with $a_i^1 < 1$, we have that $\text{MaxUE}(\mathcal{T}_i, a_i^1) > 0$ (otherwise, $\boldsymbol{a}^1$ would a perfectly secure strategy as Eve's uniform maximum payoff would be 0). Since $\text{MaxUE}(\mathcal{T}_i, a_i)$ is strictly increasing at a_i^1, this implies that $a_i^1 > a_i^2$ for these employees. However, this leads to the contradiction $1 = \sum_i a^1 > \sum_i a^2 = 1$. Therefore, the claim of the theorem has to hold. □

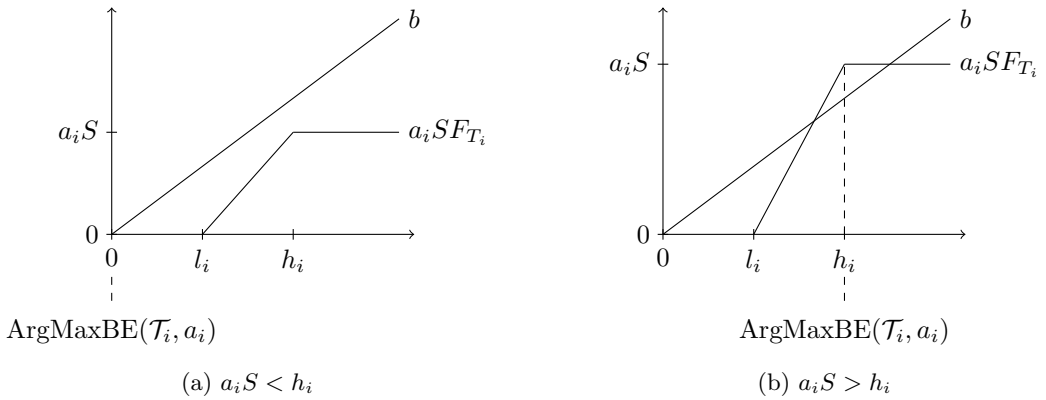
5.4 Special Case: Uniform Distributions on Trustworthiness

In this section, I study the special case of uniform trustworthiness distributions. Formally, the trustworthiness level of each employee i is assumed to be generated by a uniform random variable $T_i \sim \mathcal{U}(l_i, h_i)$, $0 < l_i < h_i < S$. In other words, employee i never reveals the secret for a bribe less than l_i, always reveals it for a bribe greater than or equal to h_i, and the probability of revealing increases linearly between l_i and h_i. Note that I allow a different distribution, i.e., different l_i and h_i, for each employee.

I begin the analysis with computing Eve's optimal bribe values for a given mixed strategy $\boldsymbol{a}$ of Alice.

Lemma 12. *Eve's optimal bribe values are*

$$\text{ArgMaxBE}(\mathcal{T}_i, a_i) = \begin{cases} \{0\} & \text{if } a_i < \frac{h_i}{S} \\ \{0, h_i\} & \text{if } a_i = \frac{h_i}{S} \\ \{h_i\} & \text{otherwise.} \end{cases} \tag{5.24}$$



(a) $a_iS < h_i$ (b) $a_iS > h_i$

Figure 5.2: Illustration for the proof of Lemma 12.

Proof. First, it is clear that no bribe value in $(0, l_i]$ can be optimal as the probability of successfully bribing is zero in this interval; thus, these bribe values are all dominated by 0. Second, it is clear that no bribe value greater than h_i can be optimal as the probability of successful bribing reaches its maximum

at h_i; thus, all values greater than h_i are dominated by h_i. For bribe values in $[l_i, h_i]$, Eve's expected payoff when targeting employee i is

$$S \cdot a_i \cdot \frac{b - l_i}{h_i - l_i} - b \ . \tag{5.25}$$

See Figure 5.2 for an illustration. When $h_i > S \cdot a_i$ (Figure 5.2a), we have that $S \cdot a_i \cdot \frac{b-l_i}{h_i-l_i} - b < S \cdot a_i \cdot \frac{b}{h_i} - b < 0$; thus, the only optimal bribe value is 0. On the other hand, when $h_i < S \cdot a_i$ (Figure 5.2b), we have that, for a bribe value $b = h_i$, the payoff is $S \cdot a_i \cdot \frac{h_i-l_i}{h_i-l_i} - h_i > 0$. It is also easy to see that the derivative of the expected payoff as a function of b is strictly greater than zero in this case; thus, the only optimal bribe value is h_i. Finally, when $h_i = S \cdot a_i$, we have that, for a bribe value $b = h_i$, the payoff is $S \cdot a_i \cdot \frac{h_i-l_i}{h_i-l_i} - h_i = 0$; thus, both 0 and h_i are optimal. □

For uniform trustworthiness level distributions, the following theorem characterizes the equilibria of the game.

Theorem 25. *If the trustworthiness level of each employee i is generated according to a uniform distribution $\mathcal{U}(l_i, h_i)$, $0 < l_i < h_i < S$, the equilibria of the game can be characterized as follows.*

- *If $k < \frac{\sum_i h_i}{S}$, then Alice is* perfectly secure*: in any equilibrium, $a_i \leq \frac{h_i}{S}$ for every i, Eve never bribes any of the employees, and both players' payoffs are zero.*
- *If $k = \frac{\sum_i h_i}{S}$, then in any equilibrium of the game, $a_i = \frac{h_i}{S}$ for every i, and Eve's payoff is zero.*
- *If $k > \frac{\sum_i h_i}{S}$, then in any equilibrium of the game, $a_i > \frac{h_i}{S}$ and $B_i \equiv h_i$ for every i, and Eve's payoff is strictly positive while Alice's payoff is strictly negative.*

Proof. Let $\boldsymbol{a}$ and $(\boldsymbol{e}, \boldsymbol{\mathcal{B}})$ be Alice's and Eve's mixed strategies, and assume that this strategy profile is a Nash equilibrium. I prove each case separately:

- $k < \frac{\sum_i h_i}{S}$: For the sake of contradiction, suppose that $a_i > \frac{h_i}{S}$ for some i. Then, there has to be a j such that $a_j < \frac{h_i}{S}$, otherwise $\sum_i a_i = k < \frac{\sum_i h_i}{S}$ would not hold. Consequently, $\text{MaxUE}(\mathcal{T}_i, a_i) > \text{MaxUE}(\mathcal{T}_j, a_j)$ and, from Lemma 11, we have that $e_j = 0$. Furthermore, from Theorems 21 and 22, we also have that $e_i > 0$. Therefore, Alice can increase her payoff by decreasing a_i and increasing a_j, which contradicts the equilibrium condition. Thus, $a_i \leq \frac{h_i}{S}$ has to hold for every i.

  Now, for the sake of contradiction, suppose that Eve targets and bribes employee i non-zero probability, that is, $e_i > 0$ and $B_i \not\equiv 0$. Since Eve's strategy has to be a best response, we have that $a_i \geq \frac{h_i}{S}$. Consequently, there has to exist some j satisfying $a_j < \frac{h_i}{S}$. From Lemma 11, we have that $e_j = 0$. Therefore, Alice can increase her payoff by decreasing a_i and increasing a_j, which contradicts the equilibrium condition. Thus, Eve never bribes any of the employees, and it follows immediately that both players' payoffs are zero.

- $k = \frac{\sum_i h_i}{S}$: For the sake of contradiction, suppose that $a_i > \frac{h_i}{S}$ for some i, which implies that there has to be a j such that $a_j < \frac{h_i}{S}$. Then, we can show that this leads to a contradiction using the same argument as in the first paragraph of the previous case. Thus, $a_i = \frac{h_i}{S}$ for every i. The rest follows readily from Lemma 12.

- $k > \frac{\sum_i h_i}{S}$: First, it is easy to see that, for any strategy $\boldsymbol{a}$, there has to be at least one i such that $a_i > \frac{h_i}{S}$, which implies $\text{MaxUE}(\mathcal{T}_i, a_i) > 0$. By using the strategy $e_i = 1$ and some constant bribe value from $\text{ArgMaxBE}(\mathcal{T}_i, a_i)$, Eve can achieve a positive payoff. Consequently, for every strategy $\boldsymbol{a}$, Eve's best response payoff has to be strictly positive. It follows immediately that, in any equilibrium, Eve's payoff is strictly positive while Alice's payoff is strictly negative.

  Now, for the sake of contradiction, assume that $a_i \leq \frac{h_i}{S}$ for some i, which implies $\text{MaxUE}(\mathcal{T}_i, a_i) = 0$. Then, we have that $e_i = 0$ from Lemma 11. Therefore, Alice can increase her payoff (i.e., decrease her loss) by increasing a_i and decreasing every non-zero component of her strategy, which contradicts the equilibrium condition. Thus, $a_i > \frac{h_i}{S}$ has to hold for every i.

  Second, assume indirectly that, for some $\boldsymbol{a}$ and $\boldsymbol{e}$ that form an equilibrium and some i, $a_i < \frac{h_i}{S}$. If $e_i = 0$, then Alice would be able to increase her payoff (i.e., decrease her loss) by simultaneously

increasing a_i and decreasing some $a_j > \frac{h_i}{S}$, which would contradict the assumption that $\boldsymbol{a}$ and $\boldsymbol{e}$ form an equilibrium. On the other hand, if $e_i > 0$, then Eve would be able to increase her payoff by simultaneously decreasing e_i and increasing e_j where j is such that $a_j > \frac{h_j}{S}$, which would also lead to a contradiction. Therefore, we have that $a_i \geq \frac{h_i}{S}$ for every i in any equilibrium. Finally, $B_i \equiv h_i$ follows readily from Lemma 12.

□

5.5 Related Work

Even though it is well-recognized in the security community that insider attacks have the potential to cause significant damage to an organization, there has been little research into developing models and techniques for analyzing the problem [Probst et al., 2006]. Due to this lack of techniques, analysts cannot assess insider threats correctly, and may consider such attacks as unpreventable [Chinchani et al., 2005]. In [Bishop and Gates, 2008], the authors point out that the "lack of a consistent definition of an insider hinders research in the detection of threats from insiders". To address this issue, they propose a definition of an insider, which can be extended across various domains and takes both cyber and physical security issues into account.

Complementary to the results presented in this chapter are models and other approaches to exhaustively find the different ways in which an insider attacker can get access to a specific resource. For example, in [Chinchani et al., 2005, Chinchani et al., 2010], the authors describe a modeling methodology which captures several aspects of insider threats and – based on the model – a threat assessment methodology to reveal possible attack strategies of an insider. As another example, the authors of [Probst et al., 2006] analyze a formal model of systems using process algebra to identify which actions may be performed by whom, which allows to compute a superset of audit results. These models typically assume that a willing insider is already in place and the obstacle is merely how to extract information from the organization. The model proposed in this chapter, on the other hand, is focused on preventing an outside attacker to successfully "turn" an insider who has knowledge of a business secret or intellectual property (and does not necessarily need to breach sophisticated access control systems to leak information to the outsider).

There are many additional research directions covering the subject of insider threats, including game theory [Liu et al., 2008] and trust models [Colwill, 2009], which are all tangent to the model proposed in this chapter. But, to the best of my knowledge, none of the published models gives directions for a project manager on how to staff a team that has to know a specific intellectual property, while being aware that an attacker might try to bribe one of his personnel. I respond to the call for research that looks "beyond information technology to the organizations overall business processes" to prevent insider threats from causing substantial harm [Cappelli et al., 2009].

This chapter also touches several other research areas. The struggle between hiders of information and seekers of information is ubiquitous in the study of steganography, the field from which the modeling idea of this chapter originated [Johnson et al., 2012, Schöttle et al., 2013, Johnson et al., 2013]. This inspiration arose from exploring the plight of a steganographer who wishes to hide k bits in a binary cover sequence of length n, and a steganalyst who wishes to detect whether the sequence has been modified. That model differs significantly from the model presented here, as the authors assume an equal a priori probability of modified and unmodified sequences, and the function that measures the predictability of sequence positions is part of the model as a parameter.

Another area that is directly connected to the situation I model is the organization of firms under weak intellectual property rights. For example, in [Rønde, 2001], the author considers a situation in which a monopolist may distribute intellectual property across two employees. There is also a competitor who might hire one of these two to gain access to the intellectual property. The author models this situation as a leader–follower game, and derives equilibria.

Proposals for deterrence strategies to prevent misuse of computing resources are complementary to the results presented in this chapter [D'Arcy et al., 2009]. These strategies may include security education and training, awareness programs, and computer monitoring. However, the effectiveness of such approaches against sophisticated insider threats is a cause for concern. A report from the intelligence community on insider threats therefore highlighted the importance of the monitoring dimension by suggesting that researchers should "focus on detection, not prevention" when fighting insider threats [Brackney and

Anderson, 2004]. The lack of the perceived ability to focus on prevention might partly rest on the lack of appropriate models and methods (beyond the basic strategies outlined above). Research on secure composition of teams addresses this problem space.

5.6 Conclusions

In this chapter, I introduced a game-theoretic model for studying the decision making of a project manager who wants to maximize the security of an organization's intellectual property. Motivated in part by known behavioral methods of assessing trustworthiness [Munshi et al., 2012], I assume that both the project manager and her adversary know the distribution of a random variable representing the trustworthiness of each employee. Finally, I assume that both players are able to estimate the value of the organization's intellectual property [Bontis, 2001].

As a result of the analysis, I find that a project manager should select every employee with a non-zero probability, unless there is a secure strategy, where an adversary has no incentives to attack at all. This contradicts the naïve assumption that, to achieve maximal security, only the most trustworthy employees should be selected. The explanation for this is the following: selecting the team members deterministically always gives the adversary the knowledge of which employees to target for bribing. So, by randomizing her strategy, the project manager minimizes the information available to the adversary for planning her attack. It is an even more surprising result that, in an equilibrium, the adversary is at most as likely to target employees that certainly know the secret as those employees that know the secret with a probability less than 1. Again, this contradicts the naïve assumption that an adversary will try to bribe the employees that are the most likely to know the secret.

For the special case of uniform distributions on trustworthiness levels, I find that the game has two distinct outcomes: either the number of team members is small enough, such that the project manager has a perfectly secure strategy, or the security of the secret depends solely on the randomness of selecting the employee with whom it is shared.[5] In the former case, the adversary has no incentives to attack and, consequently, never learns the secret. In the latter case, the adversary always attacks and always bribes the targeted employee with the minimal amount that is never below the employee's trustworthiness level. Thus, if the adversary targeted an employee that actually knows the secret, then it is certainly revealed. The project manager's only possible defense in this case is to randomize the selection of employees.

5.7 Related Publications

- Laszka, A., Johnson, B., Schöttle, P., Grossklags, J., and Böhme, R. (2013e). Managing the weakest link: A game-theoretic approach for the mitigation of insider threats. In *Proceedings of the 18th European Symposium on Research in Computer Security (ESORICS)*, pages 273–290

[5] Note that the probability that an exact equality occurs is negligible in practice.

Chapter 6

Conclusions

6.1 Summary of Results

In this section, I first summarize the main results presented in this dissertation and then highlight some of the common aspects and the differences between the results of different chapters. For a more detailed discussion of the results of a given chapter, see the conclusions at the end of that chapter. For a more application-oriented discussion of the results, see the next section (Section 6.2).

In Chapter 2,

- I showed that solving a network blocking game is an NP-hard problem in general (Theorem 2), but for a certain subclass of communication models, the game can be solved efficiently (Lemma 1 and the discussion following it).
- I proposed a novel communication model, called the All-to-One model, and showed that the game can be solved efficiently in this model (Theorem 3). I expressed the resulting robustness metric in closed-form (Theorem 5) and showed that it is equivalent to a previously proposed metric, called directed graph strength, when attack costs are assumed to be all zero.
- I proposed a novel communication model, called the All-to-All with linear usage model, and showed that the game can be solved efficiently in this model (Theorem 6). I expressed the resulting robustness metric as a graph partitioning problem (Theorem 7 and Corollary 1) and showed that it is an "extension" of a previously proposed metric, the Cheeger constant.
- I generalized network blocking games by introducing a usage-based cost model and budget constraints on the operator. I proposed two budget constraint formulations, the Maximum Cost Constraint and the Expected Cost Constraint. For three communication models that can be solved efficiently in the unconstrained game, I showed that solving the game under the Maximum Cost Constraint is an NP-hard problem (Theorem 10). Finally, I showed that the subclass of communication models that can be solved efficiently in the unconstrained game can also be solved efficiently under the Expected Cost Constraint (Theorem 11 and Lemma 5).

In Chapter 3,

- I formulated the Sink Selection with Required Persistence problem and showed that it is NP-hard (Theorem 12). To solve the problem in practice, I proposed greedy heuristic and genetic metaheuristic algorithms and demonstrated their performance using simulations (Section 3.7).
- I formulated the Sink Placement with Required Persistence problem and showed that it is NP-hard (Theorem 14). To solve the problem, I proposed an optimal search-space reduction technique and showed that it is efficient both theoretically (Theorems 15 and 16) and using simulations (Section 3.7).

In Chapter 4,

- I formulated the covert compromise scenario as a two-player game between a defender and a targeting attacker. I characterized the defender's best-response strategies (Lemma 6), the targeting attacker's best-response strategies (Lemma 8), and the game's equilibrium strategy profiles (Theorem 18).
- I formulated a sequential variant of the game and characterized its equilibria (Theorem 19). Using numerical results, I showed that the defender's payoff can be much higher in the sequential variant than in the simultaneous one (Section 4.4).

In Chapter 5,

- I formulated the bribe-resilient team composition problem as a two-player game between a manager and a spy. I gave necessary conditions on the players' best-response strategies (Lemmas 9 and 11) and equilibrium strategies (Theorems 21 and 22).
- I showed that the game always has at least one equilibrium strategy profile using a constructive proof (Theorem 23), and I proved that defender's strategy and the attacker's payoff are unique (Theorem 24 and Corollary 4).
- Finally, I characterized the game's equilibria for the special case of uniform trustworthiness level distributions (Theorem 25).

In Chapters 2, 4 and 5, I used game theory to model the conflict between a strategic attacker and the defender. To study these games, I primarily used the notion of Nash equilibrium as a solution concept. Regarding the equilibria of a game, two of the most important questions are existence and uniqueness. More specifically, these questions ask whether a game has any Nash equilibrium strategy profiles, and if it does, then whether multiple equilibria with different payoffs exist. It is easy to see that these questions are very important. First, if most instances of a problem do not have any equilibria, then results derived for the equilibrium strategies and payoffs are generally not meaningful. Second, if an instance has multiple equilibria with substantially different payoffs and behavior, then robustness assessment becomes more complicated, as we have to take all equilibria into account.

For network blocking games, we have from earlier results that a game always has at least one equilibrium and the adversary's equilibrium payoff is always unique (see Theorem 1). Obviously, these results apply to the proposed communication models as well, the All-to-One and the All-to-All with linear usage models. However, when generalizing the game to include budget constraints, I modify the basic definitions, so the results do not necessarily apply anymore. Nevertheless, it is easy to see that they still do. In the case of the Maximum Cost Constraint, the constraint is imposed on the pure strategies; hence, the game can easily be reduced to a network blocking game that is defined using the constrained set of feasible collections. In the case of the Expected Cost Constraint, I provided a more complex reduction to an unconstrained equivalent game, which basically showed that the equivalent game has essentially the same equilibria. Therefore, the existence and uniqueness results apply to the constrained games as well.

For the covert compromise mitigation game, I established that there are basically four different cases of equilibria. In the first three cases, the game has a unique Nash equilibrium. Hence, in these cases, the presented results characterize the players' payoffs and behavior very well. In the fourth case, however, the game does not have a Nash equilibrium. As discussed before, this is the case where the defender has an advantage, but the non-targeted attacks alone do not incentivize her to move frequently enough. Hence, in this case, we can either conclude with looking for other incentives to keep the defender moving, or we can switch to the sequential game formulation by advising the defender to publicly announce her strategy.

Finally, in the bribe-resilient team composition problem, I showed that the game always has at least one Nash equilibrium. Furthermore, I proved that either the manager has a perfectly secure strategy, in which case it does not matter which perfectly secure strategy she chooses, or her equilibrium strategy is unique. Consequently, the attacker's payoff is always unique.

Another important concern that is common to all the problems discussed in this dissertation is the question of computational complexity. In network blocking games and bribe-resilient team composition, the cardinality of the defender's pure-strategy set is exponential in the size of the input. Consequently,

any algorithm that uses the canonical representation for mixed defender strategies (i.e., a distribution over the set pure strategies) obviously has exponential running time. In the case of team composition, this issue could be sidestepped by using a simpler representation for the defender's mixed strategies, which is payoff-equivalent to the canonical one. In the case network blocking games, on the other hand, the problem of solving a game in general proved to be an NP-hard problem. Nevertheless, for an interesting subclass of communication models, I showed that the problem posed by the exponential size of the defender's pure-strategy set can actually be sidestepped. Furthermore, for both team composition and the proposed communication models (i.e., the All-to-One and the All-to-All with linear usage models), I showed that an equilibrium strategy profile can be computed in polynomial time from a simplified representation of the mixed strategies, given that we only have to output the support of the defender's mixed strategy.

In the sink selection problem, the cardinality of the set of possible selections is exponential in the size of the input. Unfortunately, contrary to the above game-theoretic models, it is not possible to circumvent this problem: sink selection based on maximizing persistence is indeed an NP-hard problem. Hence, to be able to solve the problem in practice, I proposed heuristic and meta-heuristic algorithms, which approximate the optimal solution reasonably well. The placement problem posed a different challenge. In this case, the cardinality of the search space is greater than continuum. After showing that this problem is also NP-hard, I reduced it to the problem of sink selection by proposing an efficient search-space reduction technique.

Finally, in the covert compromise mitigation game, both players' pure-strategy sets consist of cumulative distribution functions; hence, their cardinalities are greater than continuum. However, I showed that the attacker has only two best-response strategies, not moving and moving immediately, while the defender' best-response strategies are not moving and moving periodically. The latter consists of a continuum number of strategies (the set of possible periods is the set of positive real numbers); nevertheless, we can find the best-response to a given attacker strategy efficiently using numerical methods, given that the attack time distributions themselves are not computationally intractable.

6.2 Application of Results

Robustness of Network Topologies

In Chapter 2, I studied the robustness of network topologies against strategic attacks. More specifically, my goal was to find metrics for quantifying the robustness of network topologies, which is essential to designing attack-resilient networks. While one can find a large number of robustness metrics in the literature, most of these are not based on models of how the adversary attacks and how the network operates, or they disregard the strategic interactions between an adversary and the operator of the network. I, on the other hand, followed a game-theoretic approach, modeled the strategic interactions as a two-player game, and derived robustness metrics from the equilibrium payoffs. The advantage of this game-theoretic approach is that one can find the right metric by formulating one's assumptions on the adversary's capabilities and the constraints on the operation of the network as attacker and network models, respectively. In other words, one can find the right metric using the assumptions and constraints on a given application scenario. This makes it easier to identify and reason about which robustness metric to choose for a given application.

I proposed two novel communication models, the All-to-One model and the All-to-All with linear usage model. The robustness metric derived from the first communication model is closely related to a previously proposed robustness metric, directed graph strength. The difference between the two metrics is that, in directed graph strength, the adversary minimize the ratio between her reward and cost, while in the proposed metric, she minimizes the difference between her expected net reward and cost. Hence, the first one is more suitable for applications where one wants the effort required by an attack to be at least proportional to the caused damage, and the coefficient of this proportionality to be as high as possible. The second one is more suitable for applications where one faces an economically rational adversary.

The robustness metric derived from the All-to-All with linear usage communication model is closely related to the Cheeger constant (also called the edge expansion coefficient or the isoperimetric number). The difference between the two metrics is that the Cheeger constant considers only those attacks whose support is a minimal cut, while the proposed metric considers arbitrary attacks. Furthermore,

the proposed game-theoretic robustness metric can incorporate attack costs, which the Cheeger constant cannot. Finally, with the generalization presented in Section 2.6, both proposed metrics can also incorporate budget constraints on the network operator.

Even though the primary goal was to find robustness metrics that enable robust topology design, some of the results can be applied more directly to robust design. For example, the equilibrium adversarial strategies can be used to identify the edges that are most likely to be attacked. These critical edges are the "weakest links" in the network with respect to strategic attacks. Thus, to make a topology more robust, these are the edges that should be strengthened first.

Designing Robust Wireless Sensor Network Topologies

In Chapter 3, I studied the problem of designing robust WSN topologies, which are resilient to strategic attacks. Wireless sensor networks have many applications, including military, environmental, and health applications [Akyildiz et al., 2002]. For example, WSNs can be an integral part of military command, control, communications, computing, intelligence, surveillance, reconnaissance, and targeting (C4ISRT) systems. In many of these applications, the network is likely to be threatened by strategic attacks, against which it must be resilient. Thus, the presented results on robust topology design can be used for these applications readily.

Even though my primary goal was to study the problem of attack-resilient design, the presented results can be applied to other topology-design problems as well. Firstly, in Section 3.2.2, I showed that persistence can be used – besides quantifying robustness – to estimate the lifetime of a network. More specifically, if the energy expenditure per unit transmission is uniform over all the outgoing links of a sensor node, then the persistence of a network is proportional to the time until the batteries of the nodes run out. In other words, for networks that satisfy this constraint, network lifetime and attack-resilience measured in persistence are proportional to each other. Consequently, results based on maximizing persistence can be used for network-lifetime-maximizing design problems as well.

Secondly, the proposed search-space reduction technique can also be used more generally. Recall that the resulting set of candidate locations always includes an optimal solution if the design problem considers only the existence of links but not their lengths. Hence, the proposed search-space reduction technique can be applied to a wider-range of design problems that satisfy this constraint. In Section 3.6.3, I have listed a few examples of previous results from the literature, which could all be improved by the application of the proposed technique.

Mitigating Covert Compromises

In Chapter 4, I studied mitigation strategies against covert compromises of computing resources. The primary application of these results is finding optimal password or cryptographic-key renewal strategies. For example, many online services require – for security reasons – that users change their passwords from time to time. The timing of these mandatory changes is decided by the security policy makers of the service, who have to minimize both the security risks arising due to compromised accounts and the effort required from the users due to password changes. The results presented in Chapter 4 give a formal treatment of such trade-off problems and allow us to compute the economically optimal strategy for a defender. In the above example, the resource represents the account, the moving cost of the defender is the effort and time required to change the password (i.e., creating and remembering a new password), while the moving cost of the attacker is the effort and time required to compromise the account. Note that the model applies to a wider range of problems; for example, the resource could also model a private cryptographic key or a (virtual) machine.

As discussed in Section 4.6, one of the key implications of the presented results is that the optimal strategy for the defender is to move periodically. While this justifies the prevalent practice of periodic renewal of passwords and cryptographic keys, it contradicts the lesson learned from the very similar `FlipIt` model. This contradiction highlights the importance of finding the right modeling assumptions for a given problem. In other words, security policy makers need to carefully assess whether the assumptions of the presented model or those of the `FlipIt` model describe their situation better. For a list of differences in modeling assumptions between the two models, see Section 4.2.4.

The other key implication of the presented results is that the defender can achieve much higher payoffs in a sequential game, where she moves first and the attacker moves second. In practice, this means that the defender should not try to keep her strategy a secret but should rather publicly announce it,

allowing the attacker to play her best-response strategy. Even though this seemingly puts the attacker at advantage by letting her be perfectly informed, it can actually decrease losses by allowing the defender to make credible threats.

Secure Team Composition

In Chapter 5, I studied the problem of bribe-resilient team composition. Even though I formalized the specific problem faced by a manager who has to assemble a team of employees whom an adversary might try to bribe, the presented results apply to a much wider range of information hiding problems. Firstly, even tough I use the term "bribe" throughout the chapter, the adversary's bribing move can actually model all sorts of attacks that sidestep technical security, such as social-engineering or eavesdropping on a person. In a social-engineering attack, the adversary typically tries to gain the trust of an employee and then persuade her to reveal the secret. Alternatively, the adversary can also try to use coercion and force an employee to disclose the secret. In an eavesdropping attack, the adversary could try to compromise an employee's cell phone. Note that, even though such an attack requires compromising a technical system, it may sidestep the security of the target organization, which is potentially much higher.

Secondly, the results can be applied more generally than the problem of composing a team of employees. In fact, an "employee" can model any entity with whom information can be shared, such as a subcontractor, a computer system, an entire team, a database, or a facility. For example, the secret to be protected can be the blueprints of a new cutting-edge product, which have to be shared with a given number of facilities that will be manufacturing the product. Meanwhile, a competitor is trying to steal the blueprints by penetrating a facility, and she succeeds only if the penetration is successful and the facility actually has the blueprints. In this example, each "employee" represent a facility, and the attacker circumvents the security of the company that developed the product by penetrating a facility.

The results presented in Chapter 5 allow us to compute an economically optimal secret-sharing strategy. Moreover, they have a number of important implications regarding the problem in general. Quite interestingly, these implications contradict some very intuitive naïve ideas, such as sharing the secret with only the most trustworthy employees. In fact, the results show that the optimal strategy for the defender is to share the secret with every single employee with a non-zero probability unless she has a perfectly secure strategy. As another example, one might conjecture that an attacker will target the employees who are the most likely to know the secret; however, the results show the contrary in an equilibrium. These counter-intuitive implications regarding the problem of secret sharing show that extreme care must be taken when tackling this problem.

Bibliography

[Abbasi and Younis, 2007] Abbasi, A. A. and Younis, M. (2007). A survey on clustering algorithms for wireless sensor networks. *Computer Communications*, 30(14-15):2826–2841.

[Akkaya et al., 2007] Akkaya, K., Younis, M., and Youssef, W. (2007). Positioning of base stations in wireless sensor networks. *Communications Magazine, IEEE*, 45(4):96–102.

[Akyildiz et al., 2002] Akyildiz, I., Su, W., Sankarasubramaniam, Y., and Cayirci, E. (2002). Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422.

[Alon, 1997] Alon, N. (1997). On the edge-expansion of graphs. *Combinatorics, Probability and Computing*, 6(2):145–152.

[Alon, 1998] Alon, N. (1998). Spectral techniques in graph algorithms. In *Proceedings of the 3rd Latin American Symposium on Theoretical Informatics (LATIN)*, pages 206–215, Campinas, Brazil.

[Anderson, 2008] Anderson, R. (2008). *Security engineering - A guide to building dependable distributed systems (2nd Ed.)*. Wiley.

[Aoun and Boutaba, 2006] Aoun, B. and Boutaba, R. (2006). Clustering in WSN with latency and energy consumption constraints. *Journal of Network and Systems Management*, 14(3):415–439.

[Arkin et al., 2010] Arkin, E. M., Polishchuk, V., Efrat, A., Ramasubramanian, S., Taheri, J., Mitchell, J. S., and Sankararaman, S. (2010). Data transmission and base-station placement for optimizing network lifetime. In *Proceedings of the 6th ACM International Workshop on Foundations of Mobile Computing (DIALM-POMC)*, pages 23–32.

[Baggio, 2005] Baggio, A. (2005). Wireless sensor networks in precision agriculture. In *Proceedings of the 1st Workshop on Real-World Wireless Sensor Networks (REALWSN)*.

[Bandyopadhyay and Coyle, 2003] Bandyopadhyay, S. and Coyle, E. J. (2003). An energy efficient hierarchical clustering algorithm for wireless sensor networks. In *Proceedings of the 22nd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 3, pages 1713–1723.

[Bauer et al., 2006] Bauer, D., Broersma, H. J., and Schmeichel, E. (2006). Toughness in graphs - A survey. *Graphs and Combinatorics*, 22(1):1–35.

[Bencsath et al., 2012] Bencsath, B., Pek, G., Buttyán, L., and Felegyhazi, M. (2012). The cousins of Stuxnet: Duqu, Flame, and Gauss. *Future Internet*, 4(4):971–1003.

[Bettstetter, 2004] Bettstetter, C. (2004). On the connectivity of ad hoc networks. *Computer Journal*, 47(4):432–447.

[Bishop, 2005] Bishop, M. (2005). Position: "Insider" is relative. In *Proceedings of the 2005 Workshop on New Security Paradigms (NSPW)*, pages 77–78.

[Bishop and Gates, 2008] Bishop, M. and Gates, C. (2008). Defining the insider threat. In *Proceedings of the 4th Annual Workshop on Cyber Security and Information Intelligence Research (CSIIRW)*. ACM.

[Bogdanov et al., 2004] Bogdanov, A., Maneva, E., and Riesenfeld, S. (2004). Power-aware base station positioning for sensor networks. In *Proceedings of the 23rd Conference of the IEEE Communications Society (INFOCOM)*, volume 1.

[Bontis, 2001] Bontis, N. (2001). Assessing knowledge assets: A review of the models used to measure intellectual capital. *International Journal of Management Reviews*, 3(1):41–60.

[Bowers et al., 2012] Bowers, K., Dijk, M., Griffin, R., Juels, A., Oprea, A., Rivest, R., and Triandopoulos, N. (2012). Defending against the unknown enemy: Applying FlipIt to system security. In *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)*, pages 248–263. Springer.

[Brackney and Anderson, 2004] Brackney, R. and Anderson, R. (2004). Understanding the insider threat: Proceedings of a March 2004 Workshop. RAND Corporation, Santa Monica, CA.

[Bühler and Hein, 2009] Bühler, T. and Hein, M. (2009). Spectral clustering based on the graph p-Laplacian. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML)*, pages 81–88, Montreal, Canada.

[Capone et al., 2010] Capone, A., Cesana, M., Donno, D. D., and Filippini, I. (2010). Deploying multiple interconnected gateways in heterogeneous wireless sensor networks: An optimization approach. *Computer Communications*, 33(10):1151–1161.

[Cappelli et al., 2009] Cappelli, D., Moore, A., Trzeciak, R., and Shimeall, T. (2009). Common sense guide to prevention and detection of insider threats 3rd Edition – Version 3.1. Technical report, Software Engineering Institute, Carnegie Mellon University.

[Casey, 2003] Casey, E. (2003). Determining intent - opportunistic vs targeted attacks. *Computer Fraud & Security*, 2003(4):8–11.

[Chang and Tassiulas, 2004] Chang, J. and Tassiulas, L. (2004). Maximum lifetime routing in wireless sensor networks. *IEEE/ACM Transactions on Networking*, 12(4):609–619.

[Chen and Deng, 2006] Chen, X. and Deng, X. (2006). Settling the complexity of two-player nash equilibrium. In *Proceedings of the 47th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 261–272.

[Chinchani et al., 2010] Chinchani, R., Ha, D., Iyer, A., Ngo, H. Q., and Upadhyaya, S. (2010). Insider threat assessment: Model, analysis and tool. In *Network Security*, pages 143–174. Springer.

[Chinchani et al., 2005] Chinchani, R., Iyer, A., Ngo, H., and Upadhyaya, S. (2005). Towards a theory of insider threat assessment. In *Proceedings of the 2005 International Conference on Dependable Systems and Networks (DSN)*, pages 108–117.

[Chung, 2005] Chung, F. (2005). Laplacians and the Cheeger inequality for directed graphs. *Annals of Combinatorics*, 9(1):1–19.

[Chung, 1997] Chung, F. R. K. (1997). *Spectral graph theory*, volume 92. American Mathematical Society.

[Cobham, 1965] Cobham, A. (1965). The intrinsic computational difficulty of functions. In *Proceedings of the 1964 Congress for Logic, Methodology, and the Philosophy of Science*, pages 24–30.

[Colwill, 2009] Colwill, C. (2009). Human factors in information security: The insider threat – Who can you trust these days? *Information Security Technical Report*, 14(4):186 – 196.

[Cunningham, 1985] Cunningham, W. H. (1985). Optimal attack and reinforcement of a network. *Journal of the ACM*, 32(3):549–561.

[Czajko and Wojciechowski, 2010] Czajko, M. and Wojciechowski, J. (2010). Bi-criteria gateway placement problem in wireless sensor networks. *International Journal of Electronics and Telecommunications*, 56(3):215–222.

[Dall'Asta et al., 2006] Dall'Asta, L., Barrat, A., Barthélemy, M., and Vespignani, A. (2006). Vulnerability of weighted networks. *Journal of Statistical Mechanics*, 2006(4):P04006.

[D'Arcy et al., 2009] D'Arcy, J., Hovav, A., and Galletta, D. (2009). User awareness of security countermeasures and its impact on information systems misuse: A deterrence approach. *Information Systems Research*, 20(1):79–98.

[Dekker and Colbert, 2004] Dekker, A. H. and Colbert, B. D. (2004). Network robustness and graph topology. In *Proceedings of the 27th Australasian Conference on Computer Science (ACSC)*, pages 359–368.

[ESET Press Center, 2013] ESET Press Center (2013). ESET and Sucuri uncover Linux/Cdorked.A: The most sophisticated Apache backdoor. `http://www.eset.com/int/about/press/articles/article/eset-and-sucuri-uncover-linuxcdorkeda-apache-webserver-backdoor-the-most-sophisticated-ever-affecting-thousands-of-web-sites/`.

[Estrada, 2006] Estrada, E. (2006). Network robustness to targeted attacks. The interplay of expansibility and degree distribution. *European Physical Journal B*, 52(4):563–574.

[Finn, 2013] Finn, P. (2013). Chinese citizen sentenced in military data-theft case, *Washington Post*. `http://articles.washingtonpost.com/2013-03-25/world/38006926_1_development-of-military-technologies-information-and-technologies-chinese-citizen`.

[Fulkerson, 1971] Fulkerson, D. R. (1971). Blocking and anti-blocking pairs of polyhedra. *Mathematical Programming*, 1(1):168–194.

[Gandham et al., 2003] Gandham, S. R., Dawande, M., Prakash, R., and Venkatesan, S. (2003). Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In *Proceedings of the IEEE 2003 Global Communications Conference (GLOBECOM)*, volume 1, pages 377–381.

[Garey and Johnson, 1979] Garey, M. R. and Johnson, D. S. (1979). *Computer and intractability: A Guide to the NP-Completeness*. W. H. Freeman and Company.

[Grubesic et al., 2008] Grubesic, T. H., Matisziw, T. C., Murray, A. T., and Snediker, D. (2008). Comparative approaches for assessing network vulnerability. *International Regional Science Review*, 31(1):88–112.

[Gueye, 2011] Gueye, A. (2011). *A Game-Theoretical Approach to Communication Security*. PhD thesis, EECS Department, University of California, Berkeley.

[Gueye and Marbukh, 2012] Gueye, A. and Marbukh, V. (2012). A game-theoretic framework for network security vulnerability assessment and mitigation. In *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)*. Springer.

[Gueye et al., 2012] Gueye, A., Marbukh, V., and Walrand, J. C. (2012). Toward a metric for communication network vulnerability to attacks: A game theoretic approach. In *Proceedings of the 3rd International ICST Conference on Game Theory for Networks (GameNets)*.

[Gueye et al., 2010] Gueye, A., Walrand, J. C., and Anantharam, V. (2010). Design of network topology in an adversarial environment. In *Proceedings of the 1st Conference on Decision and Game Theory for Security (GameSec)*.

[Gueye et al., 2011] Gueye, A., Walrand, J. C., and Anantharam, V. (2011). A network topology design game: How to choose communication links in an adversarial environment? In *Proceedings of the 2nd International ICST Conference on Game Theory for Networks (GameNets)*.

[Gupta and Younis, 2003] Gupta, G. and Younis, M. (2003). Fault-tolerant clustering of wireless sensor networks. In *Proceedings of the IEEE Wireless Communications and Networking 2003 (WCNC)*, volume 3, pages 1579–1584.

[Han et al., 2007] Han, X., Cao, X., Lloyd, E. L., and Shen, C.-C. (2007). Fault-tolerant relay node placement in heterogeneous wireless sensor networks. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1667–1675.

[Herley, 2010] Herley, C. (2010). The plight of the targeted attacker in a world of scale. In *Proceedings of the 9th Workshop on the Economics of Information Security (WEIS)*.

[Holme et al., 2002] Holme, P., Kim, B. J., Yoon, C. N., and Han, S. K. (2002). Attack vulnerability of complex networks. *Physical Review E*, 65(5):056109.

[Johnson et al., 2012] Johnson, B., Schöttle, P., and Böhme, R. (2012). Where to hide the bits? In *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)*, pages 1–17. Springer.

[Johnson et al., 2013] Johnson, B., Schöttle, P., Laszka, A., Grossklags, J., and Böhme, R. (2013). Bitspotting: Detecting optimal adaptive steganography. In *Proceedings of the 12th International Workshop on Digital-Forensics and Watermarking (IWDW)*.

[Kashyap et al., 2006] Kashyap, A., Khuller, S., and Shayman, M. (2006). Relay placement for higher order connectivity in wireless sensor networks. In *Proceedings of the 25th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1–12.

[Kaspersky Lab, 2010] Kaspersky Lab (2010). Stuxnet worm: Insight from Kaspersky Lab. `http://www.kaspersky.com/about/news/virus/2010/Stuxnet_Worm_Insight_from_Kaspersky_Lab`.

[Kaspersky Lab, 2012] Kaspersky Lab (2012). Gauss. `http://www.kaspersky.com/gauss`.

[Kerkez et al., 2012] Kerkez, B., Glaser, S. D., Bales, R. C., and Meadows, M. W. (2012). Design and performance of a wireless sensor network for catchment-scale snow and soil moisture measurements. *Water Resources Research*, 48(9).

[Laszka et al., 2011] Laszka, A., Buttyán, L., and Szeszlér, D. (2011). Optimal selection of sink nodes in wireless sensor networks in adversarial environments. In *Proceedings of the 2nd IEEE International Workshop on Data Security and PrivAcy in wireless Networks (D-SPAN)*, pages 1–6.

[Laszka et al., 2013a] Laszka, A., Buttyán, L., and Szeszlér, D. (2013a). Designing robust network topologies for wireless sensor networks in adversarial environments. *Pervasive and Mobile Computing*, 9(4):546–563.

[Laszka et al., 2012a] Laszka, A., Felegyhazi, M., and Buttyán, L. (2012a). A survey of interdependent security games. Technical Report CRYSYS-TR-2012-11-15, CrySyS Lab, Budapest University of Technology and Economics.

[Laszka and Gueye, 2013a] Laszka, A. and Gueye, A. (2013a). Quantifying All-to-One network topology robustness under budget constraints. In *Proceedings of the joint Workshop on Pricing and Incentives in Networks and Systems (W-PIN+NetEcon)*. ACM.

[Laszka and Gueye, 2013b] Laszka, A. and Gueye, A. (2013b). Quantifying network topology robustness under budget constraints: General model and computational complexity. In *Proceedings of the 4th Conference on Decision and Game Theory for Security (GameSec)*, pages 154–174.

[Laszka et al., 2013b] Laszka, A., Horvath, G., Felegyhazi, M., and Buttyan, L. (2013b). FlipThem: Modeling targeted attacks with FlipIt for multiple resources. Technical report, Budapest University of Technology and Economics.

[Laszka et al., 2013c] Laszka, A., Johnson, B., and Grossklags, J. (2013c). Mitigating covert compromises: A game-theoretic model of targeted and non-targeted covert attacks. In *Proceedings of the 9th Conference on Web and Internet Economics (WINE)*, pages 319–332.

[Laszka et al., 2013d] Laszka, A., Johnson, B., and Grossklags, J. (2013d). Mitigation of targeted and non-targeted covert attacks as a timing game. In *Proceedings of the 4th Conference on Decision and Game Theory for Security (GameSec)*, pages 175–191.

[Laszka et al., 2013e] Laszka, A., Johnson, B., Schöttle, P., Grossklags, J., and Böhme, R. (2013e). Managing the weakest link: A game-theoretic approach for the mitigation of insider threats. In *Proceedings of the 18th European Symposium on Research in Computer Security (ESORICS)*, pages 273–290.

[Laszka et al., 2012b] Laszka, A., Szeszlér, D., and Buttyán, L. (2012b). Game-theoretic robustness of many-to-one networks. In *Proceedings of the 3rd International ICST Conference on Game Theory for Networks (GameNets)*, pages 88–98.

[Laszka et al., 2012c] Laszka, A., Szeszlér, D., and Buttyán, L. (2012c). Linear loss function for the network blocking game: An efficient model for measuring network robustness and link criticality. In *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)*, pages 152–170.

[Li et al., 2009] Li, J., Andrew, L. L. H., Foh, C. H., Zukerman, M., and Chen, H. H. (2009). Connectivity, coverage and placement in wireless sensor networks. *Sensors*, 9(10):7664–7693.

[Liu et al., 2008] Liu, D., Wang, X., and Camp, L. J. (2008). Game theoretic modeling and analysis of insider threats. *International Journal of Critical Infrastructure Protection*, 1:75–80.

[Mertens, 2006] Mertens, S. (2006). The easiest hard problem: Number partitioning. *Computational Complexity and Statistical Physics*, 125(2):125–139.

[Misra et al., 2008] Misra, S., Hong, S., Xue, G., and Tang, J. (2008). Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirements. In *Proceedings of the 27th IEEE International Conference on Computer Communications (INFOCOM)*, pages 281–285.

[Mohar, 1988] Mohar, B. (1988). Isoperimetric inequalities, growth, and the spectrum of graphs. *Linear Algebra and Its Applications*, 103:119–131.

[Mohar, 1989] Mohar, B. (1989). Isoperimetric numbers of graphs. *Journal of Combinatorial Theory, Series B*, 47(3):274–291.

[Munshi et al., 2012] Munshi, A., Dell, P., and Armstrong, H. (2012). Insider threat behavior factors: A comparison of theory with reported incidents. In *Proceedings of the 45th Hawaii International Conference on System Sciences (HICSS)*, pages 2402–2411.

[Muthaiah and Rosenberg, 2008] Muthaiah, S. N. and Rosenberg, C. (2008). Single gateway placement in wireless mesh networks. In *Proceedings of the 8th International IEEE Symposium on Computer Networks (ISCN)*.

[Myerson, 1991] Myerson, R. B. (1991). *Game theory: Analysis of conflict*. Harvard University Press.

[Nelder and Mead, 1965] Nelder, J. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7(4):308–313.

[Nochenson and Grossklags, 2013] Nochenson, A. and Grossklags, J. (2013). A behavioral investigation of the FlipIt game. In *Proceedings of the 12th Workshop on the Economics of Information Security (WEIS)*.

[Oyman and Ersoy, 2004] Oyman, E. I. and Ersoy, C. (2004). Multiple sink network design problem in large scale wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications 2004 (ICC)*, volume 6, pages 3663–3667.

[Pan et al., 2005] Pan, J., Cai, L., Hou, Y. T., Shi, Y., and Shen, S. X. (2005). Optimal base-station locations in two-tiered wireless sensor networks. *IEEE Transactions on Mobile Computing*, 4(5):458–473.

[Pham and Cid, 2012] Pham, V. and Cid, C. (2012). Are we compromised? Modelling security assessment games. In *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)*, pages 234–247. Springer.

[Poe and Schmitt, 2007] Poe, W. Y. and Schmitt, J. B. (2007). Minimizing the maximum delay in wireless sensor networks by intelligent sink placement. Technical Report 362/07, University of Kaiserslautern, Germany.

[Poeter, 2012] Poeter, D. (2012). Report: NotCompatible trojan attacks Android via hacked websites. `http://www.pcmag.com/article2/0,2817,2403911,00.asp`.

[Probst et al., 2006] Probst, C. W., Hansen, R. R., and Nielson, F. (2006). Where can an insider attack? In *Proceedings of the 4th International Workshop on Formal Aspects in Security and Trust (FAST)*, pages 127–142.

[Radzik, 1996] Radzik, T. (1996). Results and problems in games of timing. *Lecture Notes-Monograph Series, Statistics, Probability and Game Theory: Papers in Honor of David Blackwell*, 30:269–292.

[Radzik and Orlowski, 1982] Radzik, T. and Orlowski, K. (1982). A mixed game of timing: Investigation of strategies. *Zastosowania Matematyki*, 17(3):409–430.

[Randazzo et al., 2005] Randazzo, M., Keeney, M., Kowalski, E., Cappelli, D., and Moore, A. (2005). Insider threat study: Illicit cyber activity in the banking and finance sector. Technical Report CMU/SEI-2004-TR-021, Carnegie Mellon University.

[Raz and Safra, 1997] Raz, R. and Safra, S. (1997). A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the 29th ACM Symposium on Theory of Computing (STOC)*, pages 475–484.

[Reitter et al., 2013] Reitter, D., Grossklags, J., and Nochenson, A. (2013). Risk-seeking in a continuous game of timing. In *Proceedings of the 13th International Conference on Cognitive Modeling (ICCM)*, pages 397–403.

[Rønde, 2001] Rønde, T. (2001). Trade secrets and information sharing. *Journal of Economics and Management Strategy*, 10(3):391–417.

[Saltzer and Schroeder, 1975] Saltzer, J. and Schroeder, M. (1975). The protection of information in computer systems. *Proceedings of the IEEE*, 63(9):1278–1308.

[Sandhu and Samarati, 1994] Sandhu, R. and Samarati, P. (1994). Access control: Principle and practice. *IEEE Communications Magazine*, 32(9):40–48.

[Schöttle et al., 2013] Schöttle, P., Johnson, B., Laszka, A., Grossklags, J., and Böhme, R. (2013). A game-theoretic analysis of content-adaptive steganography with independent embedding. In *Proceedings of the 21st European Signal Processing Conference (EUSIPCO)*.

[Schwartz et al., 2011] Schwartz, G. A., Amin, S., Gueye, A., and Walrand, J. (2011). Network design game with both reliability and security failures. In *Proceedings of the 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, pages 675–681.

[Shi et al., 2006] Shi, Y., Hou, Y. T., and Efrat, A. (2006). Algorithm design for base station placement problems in sensor networks. In *Proceedings of the 3rd International Conference on Quality of Service in Heterogeneous Wired/Wireless Networks (QShine)*, page 13.

[Sindre and Opdahl, 2005] Sindre, G. and Opdahl, A. (2005). Eliciting security requirements with misuse cases. *Requirements Engineering*, 10(1):34–44.

[Skalka and Frolik, 2013] Skalka, C. and Frolik, J. (2013). Snowcloud: A complete data gathering system for snow hydrology research. In *Proceedings of the 5th Workshop on Real-World Wireless Sensor Networks (REALWSN)*, pages 3–14.

[van Dijk et al., 2013] van Dijk, M., Juels, A., Oprea, A., and Rivest, R. (2013). FlipIt: The game of "stealthy takeover". *Journal of Cryptology*, 26:655–713.

[Welsh, 2010] Welsh, M. (2010). Sensor networks for the sciences. *Communications of the ACM*, 53(11):36–39.

[Welzl, 1991] Welzl, E. (1991). Smallest enclosing disks (balls and ellipsoids). In *New Results and New Trends in Computer Science*, volume 555 of *Lecture Notes in Computer Science*, pages 359–370. Springer.

[Wong et al., 2004] Wong, J. L., Jafari, R., and Potkonjak, M. (2004). Gateway placement for latency and energy efficient data aggregation. In *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN)*, pages 490–497.

[Younis and Akkaya, 2008] Younis, M. and Akkaya, K. (2008). Strategies and techniques for node placement in wireless sensor networks: A survey. *Ad Hoc Networks*, 6(4):621–655.

[Youssef and Younis, 2007] Youssef, W. and Younis, M. (2007). Intelligent gateways placement for reduced data latency in wireless sensor networks. In *Proceedings of the IEEE International Conference on Communications 2007 (ICC)*, pages 3805–3810.

[Youssef and Younis, 2010] Youssef, W. and Younis, M. (2010). Optimized asset planning for minimizing latency in wireless sensor networks. *Wireless Networks*, 16(1):65–78.

[Zhadan, 1976] Zhadan, V. (1976). Noisy duels with arbitrary accuracy functions. *Issledovanye Operacity*, 5:156–177.

[Zhang et al., 2007] Zhang, W., Xue, G., and Misra, S. (2007). Fault-tolerant relay node placement in wireless sensor networks: Problems and algorithms. In *Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM)*, pages 1649–1657.