Static & Dynamic Game Theory:
Foundations & Applications

Jeffrey Pawlick
Quanyan Zhu

Game Theory for Cyber Deception

From Theory to Applications

Birkhäuser

Static & Dynamic Game Theory: Foundations & Applications

Jeffrey Pawlick · Quanyan Zhu

Game Theory for Cyber Deception

From Theory to Applications

Birkhäuser

Jeffrey Pawlick
Tandon School of Engineering
New York University
Brooklyn, NY, USA

Quanyan Zhu
Tandon School of Engineering
New York University
Brooklyn, NY, USA

ISSN 2363-8516 ISSN 2363-8524 (electronic)
Static & Dynamic Game Theory: Foundations & Applications
ISBN 978-3-030-66064-2 ISBN 978-3-030-66065-9 (eBook)
https://doi.org/10.1007/978-3-030-66065-9

Mathematics Subject Classification: 91A10, 91A28, 91A80

This book is published under the imprint Birkhäuser, www.birkhauser-science.com by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

To our parents

Preface

Interactions both online and in the emerging Internet of things (IoT) feature distributed agents with local decision-making capabilities and misaligned incentives. This has motivated the analysis of cybersecurity using game theory: a systems science that models strategic interactions between rational and self-interested parties. More specifically, cyber deception is characterized by the strategic manipulation of information asymmetry and imperfect detection. Deception plays a critical role in countless cybersecurity problems, but involves a common set of features across disciplines.

Holistic approaches to deception are needed in order to enable scientific conceptualization, modeling, analysis, and design. Vendors of cyber insurance, technical entrepreneurs, and government policy-makers all require a systematic understanding of cyber deception. Game theory provides such an analysis through equilibrium characterization and comparative statics. It also yields design methodologies that manipulate the parameters of a game in order to achieve the desired equilibrium.

This book aims to address emerging problems in cybersecurity using rich game-theoretic models. A model called signaling games with evidence is used to capture the leaky deception featured in botnet deployment. We also study broadcast deception (a phenomenon common to botnet-recruitment, reconnaissance that precedes advanced persistent threats, and deceptive opinion spam) using innovations in large-population games such as Poisson signaling games and mean-field Stackelberg games. Since attackers often execute attacks over long periods of time (e.g., through a cyber kill chain), the book develops dynamic approaches to defensive deception. Finally, inspired by engineering approaches that break down large problems into individual components, we analyze games-of-games models for cyber deception that consist of multiple games coupled together through best-response dynamics.

These models are used to address emerging challenges in cybersecurity. For example, we study physical denial-of-service attacks carried out by IoT devices on critical infrastructure, strategic trust in autonomous vehicle networks, and information privacy achieved through strategic obfuscation. In this sense, the book can

be used as a foundation for modeling and mechanism design in key areas of cybersecurity.

This work should be appropriate for graduate and advanced undergraduate students, as well as researchers in fields ranging from business to engineering. The book can also serve as an introduction to systems science for industry and government professionals in cybersecurity.

New York, USA
November 2020

Jeffrey Pawlick
Quanyan Zhu

Contents

Notation

This book includes a wide breadth of models and applications. Therefore, the mathematical notation is flexible. Nevertheless, we present some consistent styles here.

Double-struck notation (*e.g.*, $\mathbb{R}$) indicates spaces or sets. Script (*e.g.*, $\mathcal{W}$) indicates random variables. Bold (*e.g.*, $\mathbf{G}$) defines a game.

The capital letters A, D, S, R, L, and F indicate the following players: attacker, defender, sender, receiver, leader, and follower, respectively. We denote the set of players in a game by $\mathbb{PL}$. The notation $\theta \in \Theta$ indicates a sender type. This book sometimes considers heterogeneous receivers, and $\phi \in \Phi$ indicates a receiver type[1]. $\mathbb{A}$ represents a set of actions, and $\mathbb{M}$ represents a set of messages. In this book, we consider a concept called evidence, and we denote sets of possible evidence by[2] $\mathbb{EV}$. Mixed strategies are always indicated by σ. For example, in Chap. 6, $\sigma^S(m|\theta)$ indicates the mixed-strategy probability with which a sender S of type θ plays message m.

We use a lower-case u to indicate a pure-strategy utility function. An upper-case U denotes a mixed-strategy utility function. Some interactions in this book are dynamic. We use J to denote multistage utility (or cost)[3].

Other notations apply only within the chapter in which they are defined.

[1]Note that Θ and Φ define sets even though they are not double-struck.

[2]We reserve $\mathbb{E}$ and $\mathbb{P}$ for expected value and probability, respectively.

[3]There is one exception: in Chap. 6, which considers static problems, J is a scalar quantity called Youden's J Statistic.

Part I
Fundamentals

Chapter 1
Introduction

1.1 Cybersecurity

In 2008, the National Academy of Engineering (NAE) included among its 14 *Engineering Grand Challenges* the objective to "secure cyberspace" [1]. Since that time, increased awareness of threats to cyberspace has helped spur private and public investment in cyberdefense. Yet in the past 10 years, cyberthreats have not only continued to exist but have also developed new forms based on the evolution of new technologies.

In the realm of information security, attackers successfully breached the information systems of Home Depot in 2014, insurance company Anthem Inc, in February of 2015, and the US Office of Personnel Management later that year. Indeed, the 10 years from 2005 to 2015 featured at least 4,000 known data breaches [2].

Recent cyberattacks have also manifested physical consequences. The power grid in Ukraine, Iranian nuclear centrifuges, and an American water dam 20 miles north of New York were all infiltrated [3]. Many of these breaches made use of advanced persistent threats (APTs): stealthy attacks that employ social engineering and deception to give adversaries insider access to networked systems.

These security concerns have also been accompanied by worries about privacy. Online tracking and wearable computing have added complexity to data ecosystems and increased the depth to which technology operators can learn about their users [4]. In addition, the pervasiveness of tracking allows learners to infer habits and physical conditions over time. For instance, tracking algorithms may predict "a user's mood; stress levels; personality type; bipolar disorder; demographics" [5]. These are unprecedented degrees of access to user information. In summary, cybersecurity and privacy face just as many threats today as they did 10 years ago. This book can be seen as an attempt to confront that reality.

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_1

1.1.1 The Internet of Things

The past 10 years have also seen significant development in the so-called Internet of things (IoT). While data breaches, attacks on critical infrastructure, and threats to privacy can all be carried out over traditional information networks, their breadth and depth will be increased by further development in the IoT. In this book, we therefore focus especially on security and privacy challenges in the IoT. Here, we introduce the technological components of the IoT, its relation to the control of physical systems, and the features of the IoT that give rise to challenges in security and privacy.

A European Commission in 2010 defined the IoT as a "dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols where physical and virtual 'things' have identities, physical attributes, and virtual personalities" [6]. Many different visions have been proposed for the IoT. Here, we focus on three salient paradigms: *network-centric*, *cloud-centric*, and *data-centric* [7]. These paradigms are not mutually exclusive. But they feature different goals and requirements.

Network-Centric IoT

The name IoT apparently originated with *The Auto-ID Labs* [8], a network of research laboratories involved in network radio-frequency identification (RFID) research and emerging sensing technologies [9]. A network-centric IoT grows out of developments in passive RFID tags, wireless sensor networks (WSN), crowd sensing, and social networks [7]. Each of these domains consists of decentralized networks made up of multiple entities.

Note that these entities have increasingly *active* capabilities. Passive RFID tags have no power source of their own and are mainly used for tracking [10]. WSN are distributed data-gathering devices that run firmware or simple operating systems. They can automatically update their configurations and make simple decisions. Crowd sensing involves an even larger degree of agency. In this arena, humans decide what data is gathered and transmitted [11]. These decisions may involve differing incentives and strategic competition. Finally, in social networks, even the topology of the network is determined by humans [12]. Users frequently break or create their own links. In summary, the network-centric vision of the IoT consists of a network of interacting agents that have possibly misaligned incentives.

Cloud-Centric IoT

The cloud-centric IoT emphasizes the availability of centralized computational resources, data, and software. RFID and WSN can involve thousands of nodes with tight space and power constraints. This motivates the need to offload data and processing to a remote resource such as a cloud. Remote resources may offer Infrastructure, Platforms, and Software-as-a-Service (IaaS, PaaS, and SaaS) [13, 14]. In exchange for providing IoT devices with infrastructure, platforms, and software, vendors obtain a centralized market for their services in the cloud [7].

Clouds are centralized in the sense that they are accessed by multiple devices and operated by a single agent. On the other hand, new paradigms allow resources such

as a cloud to be more distributed. For example, *fog computing* envisions moving computational, storage, and networking services from a centralized cloud further toward the edge of the network. Distributed agents in a fog form a middle layer between sensors, actuators, and the higher cloud or data center [15]. Like cloud computing, fog computing will continue to offer IoT devices access to resources not available locally.

Data-Centric IoT

The data-centric IoT emphasizes the role of computational intelligence, machine learning, and data-to-information analytics. In this paradigm, data comes from both people and "things." The data must be processed in order to obtain useful information. Often it must then be made available to humans through interpretation and visualization [7]. One scholar breaks these processes into four stages: (1) *data acquisition*, (2) *information processing*, (3) *meaning-making*, and (4) *action-taking* [16]. Data is transformed through these stages into actionable information.

New IoT devices have made possible tremendous amounts of data gathering. Smart watches, wearable medical sensors (for monitoring heart rate, glucose level, and possibly kidney function and electrolyte balance), home automation systems, and environmental monitors provide continuous data streams scaling from intensely personal to broadly distributed [16]. Of course, some of this data can be used to infer sensitive individual information such as sleep and exercise habits, activity in the home, and travel patterns. Hence the data-centric IoT paradigm also motivates privacy concerns. Some technology is already being developed to obfuscate data collected through the IoT in order to protect users' privacy [17, 18].

1.1.2 The IoT and Cyber-Physical Systems

Importantly, the IoT does not merely gather and process information from the environment. The IoT also makes it possible to act on that information in order to change the environment. Specifically, the IoT may also include automatic actuators. The following passage refers specifically to *sensor-actuator networks*, but many of the principles which it enumerates are relevant to what is called the Internet of *Controlled* Things (IoCT).

> Sensor-actuator networks are heterogeneous networks that comprise networked sensor and actuator nodes that communicate among each other using wireless links to perform distributed sensing and actuation tasks. Actuators (called also actors) are resource-rich, potentially mobile, and are involved in taking decisions and performing appropriate actions on themselves (e.g. controlled movement), on sensors (such as activating sensors, moving or replacing a sensor), and/or in the environment (e.g. turn on their own... water sprinkler to stop the fire). Sensor-actuator networks are expected to operate autonomously in unattended environments. They may be directly connected (using, for instance, web infrastructure) and responsive to a user (task manager) who controls the network via sinks [19].

In this vision, humans are only one type of actuator in the IoT. Rather, connected "things" act on data and also act on themselves. Thus arises the term Internet of

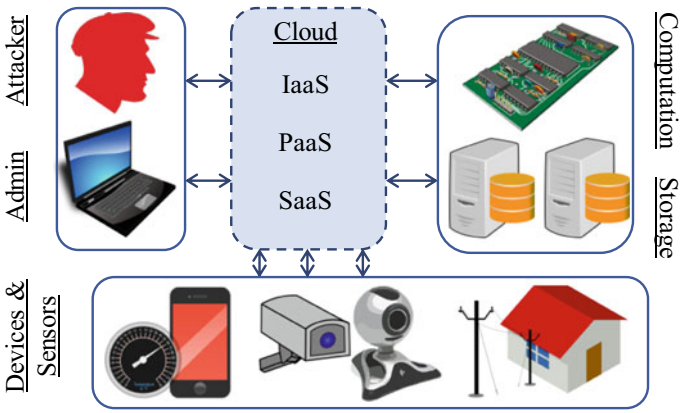


Fig. 1.1 The IoCT consists of connected sensors and devices, possibly with a cloud as the interface. Adversaries may be capable of compromising cloud services and modifying the control signals that they transmit to the devices

Controlled Things. This term emphasizes that "things" are part of the feedback loop. Control systems, located either locally or in the cloud, are responsible for optimally processing sensed data and providing control signals to the "things." Figure 1.1 depicts an example of an IoCT in a smart home. The term IoCT was mentioned in a 2008 presentation for the NSF, but the next earliest use seems to have been within our laboratory [20].

1.1.3 Broad Features of the IoT/IoCT

Each of these paradigms—network-centric, cloud-centric, and data-centric IoT, together with the idea of the IoCT—suggests broad features that we consider in this book. These features motivate our analysis of the IoT and our design of mechanisms to improve it.

Plug-n-Play Functionality

The network-centric vision requires diverse agents to be capable of seamlessly interacting over a common protocol. The IoT should allow agents to easily enter and leave without the necessity of totally reconfiguring the network. Devices should also be able to utilize network resources without prior knowledge of the entire set of connected devices. In summary, the IoT should support *plug-n-play* functionality. In the security domain, this adds difficulty in obtaining knowledge about adversaries and to attributing cyberattacks to their perpetrators.

Strategic Interaction

Devices which communicate with each other or with a centralized resource such as a cloud must judge whether the other party is trustworthy. A lack of trust could be due to multiple factors. First, agents could simply have misaligned incentives; different devices competing for bandwidth, energy, etc., could have incentives to act deceptively or at least strategically. Second, some actors in the IoT could be actively malicious. Attackers interested in damaging devices or disrupting functionality could transmit malicious signals. In the face of deception, agents require intelligent means of deciding whether to trust other network components.

Ubiquity

The data-centric vision of the IoT emphasizes the ubiquitous collection of data. Indeed, the presence of localized data-collecting devices and data-forwarding devices (smartphones, tablets) means that many environments constantly emit data. This ubiquity suggests that the IoT is a relevant concept for applications ranging from personal health to smart homes and cities. It also suggests the need for privacy and highlights recent developments in obfuscation and defensive deception.

Cyber-Physical Systems

With the term IoT, we have emphasized the presence not only of sensors and data but also of actuators and physical devices. Control of things in the IoT requires knowledge of the physical systems involved. Often these systems have critical features that must be protected. For instance, glucose control systems must be aware of the dynamics of insulin and blood sugar, and pacemakers must be programmed with an understanding of which settings are normal and which could result in loss of life [21]. The term *cyber-physical systems* (CPS) has been used to emphasize the interaction of both cyber and physical components [22]. While the multi-layer nature of CPS creates a large attack surface, it also gives rise to multi-layer security strategies. Besides allocating security resources at the cyber-layer, designers can also use robust control policies and local control backups to implement *defense-in-depth* [23, 24].

Dynamics

Finally, the IoT is dynamic in three senses. In the first sense, devices can enter and leave the network, and the cyber-layer must be capable of adapting. In the second sense, the physical-layer devices interact with their environment over time, in a dynamic manner. These interactions may take place with little human intervention. This motivates the need for automatic feedback control. Finally, in the third sense, the IoT is dynamic because events in the physical layer and the cyber layer may occur at distinct times. Moreover, different agents distributed throughout the IoT may have access to information and may be able to observe events at different times. This type of dynamism is addressed by dynamic game theory models (see [25–28]).

1.2 Deception

Clearly, a recurring problem in the IoT is deception. Malicious deception is a feature of many cyberattacks, including phishing, APTs, man-in-the-middle attacks, deployment of Sybil nodes in social networks, and many others. At the same time, deception can be used by defenders to hide private information or disguise defensive techniques, tools, and procedures. More generally, deception is commonplace in adversarial or strategic interactions in which one party possesses information unknown to the other. While this book studies deception in cybersecurity, we are also motivated by challenges in deception that originate in psychology, criminology, economics, and behavioral sciences. We give a sample of these challenges here.

1.2.1 Deception Across Disciplines

Military Applications

Deception and secrecy demanded significant attention during World War II and the Cold War [29, 30]. But increasing globalization and the proliferation of communication technologies have recently created further challenges for mitigating deception. The increasing availability of information has led not only to more knowledge but also to worse confusion [31]. National defense requires a detailed study of military-relevant deceptions such as APTs carried out by state actors [32].

Psychology and Criminology

Research in psychology and criminology suggests that humans have poor abilities to detect deception [33, 34]. One approach to address this shortcoming focuses on interview techniques. It has been shown that detection rates can be improved by tools that increase the cognitive load by asking suspects to recall events in reverse order, to maintain eye contact, or to answer unexpected questions. Some of these have been incorporated into the investigative protocol known as the *Cognitive Interview for Suspects* (CIS) [35]. A second approach uses physiological indicators. For instance, the *Guilty Knowledge Test* (GKT) prompts a suspect with a list of items—for example, a set of articles found at the scene of a crime—and measures the suspect's physiological responses to each item [36]. Signs of arousal in a suspect suggest that the suspect possesses guilty knowledge, because the articles are irrelevant to an innocent person.

Privacy Advocacy

Recently, privacy advocates have designed technologies for Internet users to *obfuscate* the trails of their digital activity against ubiquitous tracking. Privacy advocates argue that developments such as third-party tracking and persistent cookies have not been sufficiently regulated by law. Therefore, there is a need for user-side technologies to provide proactive privacy protection. One example is *TrackMeNot*, a

browser extension that periodically issues random search queries in order to undermine search engine tracking algorithms [37]. Another example is *CacheCloak*, which protects location privacy by retrieving location-based services on multiple possible paths of a user. An adversary tracking the requests is not able to infer the actual user location [17]. These are instances of deception that is designed for benign purposes.

Behavioral Economics

In economics, the area of *strategic communication* quantifies the amount of information that can be transmitted between two parties when communication is unverifiable [38, 39]. Communication can be evaluated both strategically and physiologically. One recent paper analyzes patterns of eye movement and pupil dilation during strategic deception [40]. At the same time, research in behavioral economics finds that sometimes economic agents choose not to deceive, even when it is incentive-compatible [41–43]. Subjects that exhibit so-called *lying aversion* choose to maximize their payoffs less frequently when it requires a lie than when it requires a simple choice. This points towards the influences of morality and ethics on deception.

Economic Markets

In the broader economics literature, Akerlof and Shiller argue that many markets admit equilibria in which deceivers exploit those who are vulnerable [44]. The authors describe these interactions in politics, pharmaceuticals, finance, and advertising. They use email phishing as an analogy for any kind of deception in which an informed "phisherman" exploits the lack of knowledge or the psychological vulnerabilities of a group of "phools." The essential insight is that opportunities for deception will be exploited in equilibrium. Across all six disciplines, deception involves interaction, conflict, rationality, and uncertainty.

1.2.2 Defensive Deception in Cybersecurity and Privacy

The security landscape of the IoT facilitates deception, because information can lack permanence, attribution is difficult [45], and some agents lack repeated interactions [46]. Although firewalls, cryptography, and role-based access control are essential components of any security strategy, they are unable to fully address new cybersecurity and privacy threats. Adversaries often gain undetected, insider access to network systems. They obtain information about networks through reconnaissance, while defenders lack an understanding of the threats that they face.

New techniques, however, allow defenders to gain the upper hand in information asymmetry. The U.S. Department of Defense has defined *active cyber defense* as "synchronized, real-time capability to discover, detect, analyze, and mitigate threats and vulnerabilities... using sensors, software, and intelligence..." [47]. These techniques both investigate and deceive attackers [48]. Examples of defensive deception include moving target defense [49], honeypot deployment [50], and the use of mix networks [51]. Chapter 4 gives a definition of deception and presents a taxonomy that

includes six types of defensive deception. Chapters 5–7 study three of these species in depth. Chapters 8 and 9 study mitigation of malicious deception.

1.3 Systems Sciences

Much research in the IoT so far has focused on specific application domains. These domains include, for instance, the smart grid, mobile and vehicular ad-hoc networks (MANET and VANET), cloud computing, and consumer electronics. On the other hand, there is a lack of a systematic understanding of the IoT which would enable the design of holistic solutions. Certainly, existing lines of research consider the IoT holistically in terms of architectures and conceptual frameworks. We have drawn from these works, (e.g., [6, 7, 52, 53]) in the above outline of the architectures and broad features of the IoT. On the other hand, these works are typically qualitative rather than quantitative. We aim at a broad understanding of the IoT which is also mathematical.

1.3.1 Systems Science Methodology

Systems science bridges an important gap between specific technologies and holistic understandings. Systems science leans toward a philosophical approach, in the sense that it is motivated by concrete engineering challenges, but tries to get at the conceptual roots behind these challenges and design holistic solutions to them. Convex optimization, for instance, can be applied to many problems and outlasts any specific technologies that may utilize it. In this book, we leverage tools from many systems sciences, including signal processing, machine learning, detection and estimation, and—especially—optimal control and game theory.

Using these systems sciences, we attempt to identify philosophical properties that are *emergent*. The properties of systems such as the IoT are more than the sum of the properties of their parts. Indeed, Chap. 8 studies a type of equilibrium that we call *Gestalt Nash equilibrium* [54–56]; Gestalt refers to a psychological phenomenon that is more than the sum of its parts. In general, this book investigates issues that arise because of the confluence of multiple properties—the combination of plug-n-play, strategic, ubiquitous, cyber-physical, and dynamic properties of the IoT.

1.3.2 Applications of Systems Sciences

Systems sciences are important for several applications in economics, business, policy, and engineering. First, systems sciences allow prediction. Control-theoretic and game-theoretic models can be used to predict the equilibrium implications of changes

in specific variables that characterize a system. This prediction is relevant for government and corporate policy. For instance, lawmakers need to understand what size penalty is necessary to dissuade someone from running a phishing scheme or selling stolen credit card numbers. Prediction is also needed for a bank to decide when to reissue credit cards if some of the accounts may have been compromised [57].

Systems sciences also enable *mechanism design*. In the context of strategic systems, mechanism design employs the reverse perspective from game theory. Game-theoretic models take the parameters of interaction as inputs, and produce a prediction of the equilibrium outcome as outputs. Mechanism design takes the desired equilibrium result as an input, and produces the parameters necessary to obtain that result as an output. More broadly, mechanism design can include changing the characteristics of the game itself. In economics, for instance, mechanism design is used to design the rules of an auction such that agents are motivated to bid truthfully. We employ mechanism design to detect and mitigate deception, as well as to improve cyber-layer security by including physical-layer defenses that shift incentives in the cyber layer.

1.4 Game Theory

Game theory "can be defined as the study of mathematical models of conflict and cooperation between intelligent rational decision-makers" [58]. It is the systems science that studies interactions between rational and strategic agents, or *players*. Agents are rational in the sense that they choose actions to optimize their own benefits (or *utility*). But their utility functions depend both on their own actions, and on the actions of the other agents. In game theory, the agents are strategic in the sense that they choose their own actions by anticipating the actions of the other agents. One way to summarize this is to say that game-theoretic agents optimize coupled utility functions.

Game-theoretic utility functions model varying interaction contexts. Additionally, game theory includes tools that capture dynamics, information asymmetry, and randomized actions. This systems science has been used in a wide variety of security and cybersecurity contexts. A few application areas include intrusion detection systems [59–61], adversarial machine learning [62–65], communications channel jamming [66–69], critical infrastructure protection [70–73], and industrial control systems [27, 74–77]. Practical applications in physical security include the ARMOR system for Los Angeles Airport Security [78] and the PROTECT system for patrol routes at the port of Boston [79]. In each of these contexts, attackers and defenders attempt to rationally optimize their actions by anticipating the actions of the other players. Hence, game theory captures the fundamental properties of these interactions. Chapters 2–3 further introduce game theory its fundamental mathematical models.

1.5 Outline of the Book

Chapters 2 and 3 lay the theoretical foundations for the rest of the book. These chapters introduce fundamental game-theoretic models such as Nash, Stackelberg, and signaling games. They also give a sampling of paradigmatic games, (e.g., market entry, matching pennies) and an idea of problems that arise in game theory, such as equilibrium selection. Readers with a strong background in game theory can go directly to Chap. 4.

Chapter 4 provides a high-level overview of defensive deception. This chapter is critical for understanding the book. It begins with a brief glance at 25 research works in defensive deception published during the years 2009–2019, i.e., the 10 years since the announcement of NAE's *Engineering Grand Challenges*. The chapter continues by clustering the deception techniques used in these works into six different categories or species. In the literature, each of these species is modeled using one or more game-theoretic approaches. We argue for the modeling approaches that we think are most promising, and we use these models in the rest of the book.

Chapters 5–7 study three of the six deception species: obfuscation, honey-x, and attacker engagement. These are very different types of deception, and thus they require very different theoretical approaches. Through obfuscation, a defender attempts to disguise valuable information by adding noise. Therefore, models of obfuscation require concepts drawn from information theory and machine learning. In honey-x, a defender attempts to lead an attacker into a trap by sending a deceptive signal. This motivates the use of signaling games. Finally, in attacker engagement, a defender deceives an attacker over an extended period of time. Hence, attacker engagement is modeled using Markov decision processes. Obviously, then, the theory in these chapters is not cumulative. Rather, each chapter introduces the reader to a different approach to studying deception.

Chapters 8–9 shift the perspective and ask how a defender can mitigate deceptive techniques used by an attacker. These chapters build upon the concepts used in Chaps. 5–7, such as signaling games and large-population games. They also introduce new concepts such as Gestalt Nash equilibrium (Chap. 8) and Poisson signaling games (Chap. 9). They study emerging challenges such as autonomous vehicle security and defense of the smart grid.

Finally, Chaps. 10–11 mention theoretical and practical gaps in cyber deception. It is clear that much work remains to be done, and we hope that these chapters can help provide inspiration for future studies and field applications. In general, our purpose in this book is not to provide definitive solutions, but rather frameworks that can be modified and built-upon by other scholars.

Chapter 2
Nash and Stackelberg Games

In this chapter, we give an introduction to several game-theoretic solution concepts that will be used in this book. The chapter starts by introducing matrix-form strategic games and the concept of Nash equilibrium. We then present extensive-form games and the concept of information sets. Stackelberg games are an important type of extensive-form games. This chapter introduces the structure of the game and the solution concept of Stackelberg equilibrium.

2.1 Zero-Sum Matrix Games

Two-person zero-sum games are strategic-form games that can be represented by a matrix. Consider two players, P1 (row player) and P2 (column player), and an $m \times n$-dimensional matrix $A = \{a_{i,j}\}$. Each entry $a_{i,j}$ of the matrix A represents the outcome of the game if P1 picks action i (i.e., the i_{th} row) and P2 picks action j (i.e., the j_{th} column). If $a_{i,j}$ is positive, P1 pays $a_{i,j}$ to P2; if $a_{i,j}$ is negative, P2 pays $|a_{i,j}|$ to P1. As a rational decision maker, P1 aims to minimize the outcome of the game, while P2 seeks to maximize it.

2.1.1 Pure Strategies

One rational strategy of P1 is to secure his losses against any behavior of P2. Hence, P1 chooses row i^*, whose largest entry is no bigger than the largest energy of any other row. If P1 chooses the i^*-th row, i^* satisfies:

$$\overline{V} \triangleq \max_j a_{i^*,j} \leq \max_j a_{i,j}, \quad i = 1, 2, \ldots, m, \tag{2.1}$$

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_2

where $\overline{V}$ is called the *loss ceiling* of P1.[1] The strategy "row i^*" is called a *security strategy* for P1.

Similarly, P2 can choose to secure his gains against any behavior of P1. To do this, P2 picks column j^*, whose smallest entry is greater than the smallest entry of any other row. In other words, j^* satisfies

$$\underline{V} \triangleq \min_i a_{i,j^*} \geq \min_i a_{i,j}, \quad j = 1, 2, \ldots, n, \tag{2.2}$$

where $\underline{V}$ is called the *gain floor* of P2.[2] The strategy "column j^*" is called a *security strategy* for P2.

Suppose now that P1 is required to announce his choice before P2 makes his choice. In this case, P2 has an advantage over P1. Then the best play of P1 is to choose one of his security strategies (say row i^*) to secure the loss ceiling. Therefore, P2's "optimal" choice (column j°) would satisfy

$$a_{i^*,j^\circ} = \max_j a_{i*,j} = \overline{V} = \min_i \max_j a_{i,j}, \tag{2.3}$$

where the "min-max" operator designates the order of play in this decision process: P1 minimizes his loss ceiling, and then P2 maximizes his gain. Equation (2.3) implies that the outcome of the game is equal to the upper value $\overline{V}$ when P2 observes P1's choice. Hence the *minimax strategy* is to minimize one's own maximum loss.

Now, if we exchange the order of two players, then P1 has an advantage over P2. The best play of P2 is to choose one of his security strategies (say column j^*) to secure the gain floor. Then P1's "optimal" choice (column i°) would satisfy

$$a_{i^\circ,j^*} = \min_i a_{i,j^*} = \underline{V} = \max_j \min_i a_{i,j}, \tag{2.4}$$

where the "max-min" symbol implies that the minimizer acts after the maximizer. Equation (2.4) indicates that the outcome of the game is equal to the lower value $\underline{V}$ when P1 observes P2's choice. So, the *maximin strategy* is to maximize one's own minimum gain.

For some matrix form, zero-sum games, the security strategies of the two players coincide. In that case, we have a *saddle-point equilibrium*.

Definition 2.1 (*Saddle-point equilibrium*) The pair (i^*, j^*) is a saddle-point equilibrium if $a_{i^*,j} \leq a_{i^*,j^*} \leq a_{i,j^*}$ is satisfied for all $i = 1, 2, \ldots, m$ and $j = 1, 2, \ldots, n$.

[1] This quantity is sometimes also called the *security level* of P1 or the *upper value* of the game.

[2] Similarly, this quantity can be called the *security level* of P2 or the *lower value* of the game.

2.1.2 Mixed-Strategies

In the discussion above, both players selected one of their possible actions deterministically. That is, both players played *pure strategies*. The existence of a saddle point in pure strategies, however, is not always guaranteed. For the case in which the security levels of the players do not coincide, no such equilibrium solution can be found within the class of pure strategies. Hence, there is a need to extend the strategy space from pure strategies to mixed strategies.

A mixed strategy for a player is a probability distribution on the space of his pure strategies. We denote the strategies of P1 and P2 as $\sigma_1 \in \Sigma_1$ and $\sigma_2 \in \Sigma_2$, respectively, where Σ_1 and Σ_2 are defined as follows:

$$\Sigma_1 := \{\sigma_1 = (\sigma_1^1, \ldots, \sigma_1^m)^T : \sigma_1^i \geq 0, \text{ for } i = 1, \ldots, m, \text{ and } \sum_{i=1}^{m} \sigma_1^i = 1\}$$

$$\Sigma_2 := \{\sigma_2 = (\sigma_2^1, \ldots, \sigma_2^n)^T : \sigma_2^i \geq 0, \text{ for } i = 1, \ldots, n, \text{ and } \sum_{i=1}^{n} \sigma_2^i = 1\}$$

Note that pure strategies can be viewed as a special case of mixed strategies. For zero-sum games with mixed strategies, the expected utility is given by $U(\sigma_1, \sigma_2) = \sigma_1^T A \sigma_2$. P1 seeks to minimize the average payoff $U(\sigma_1, \sigma_2)$ by choosing a probability distribution vector $\sigma_1 \in \Sigma_1$, while P2 aims to maximize $U(\sigma_1, \sigma_2)$ by choosing a probability distribution vector $\sigma_2 \in \Sigma_2$.

Definition 2.2 (*Mixed security strategy*) A probability distribution vector $\sigma_1^* \in \Sigma_1$ is called a *mixed security strategy* for P1 in the matrix game A if the following inequality holds:

$$\overline{V}_m(A) \triangleq \max_{\sigma_2 \in \Sigma_2} \sigma_1^{*T} A \sigma_2 \leq \max_{\sigma_2 \in \Sigma_2} \sigma_1^T A \sigma_2, \text{ for all } \sigma_1 \in \Sigma_1. \tag{2.5}$$

Here, $\overline{V}_m(A)$ is the *average security level* of P1 (or the *average upper level* of the game). Analogously, a probability distribution vector $\sigma_2^* \in \Sigma_2$ is called a mixed security strategy for P2 in the matrix game A if the following inequality holds:

$$\underline{V}_m(A) \triangleq \min_{\sigma_1 \in \Sigma_1} \sigma_1^T A \sigma_2^* \geq \min_{\sigma_1 \in \Sigma_1} \sigma_1^T A \sigma_2, \text{ for all } \sigma_2 \in \Sigma_2. \tag{2.6}$$

Here, $\underline{V}_m(A)$ is the average security level of P2 (or the average lower level of the game).

The concept of saddle-point equilibrium in Definition 2.1 can be extended for mixed strategies through Definition 2.3.

Definition 2.3 (*Saddle-point equilibrium in mixed strategies*) A pair of strategies (σ_1^*, σ_2^*) is said to constitute a saddle point in mixed strategies, if

$$\sigma_1^{*T} A\sigma_2 \leq \sigma_1^{*T} A\sigma_2^* \leq \sigma_1^T A\sigma_2^*, \text{ for all } \sigma_1 \in \Sigma_1,\ \sigma_2 \in \Sigma_2. \tag{2.7}$$

Here, $V_m(A) = \sigma_1^{*T} A\sigma_2^*$ is known as the *saddle-point value*, or the value of the game in mixed strategies.

In a mixed-strategy matrix game, the upper value and the lower value of the game are given, respectively, by

$$\overline{V}_m(A) = \min_{\sigma_1} \max_{\sigma_2} \sigma_1^T A\sigma_2, \tag{2.8}$$

$$\underline{V}_m(A) = \max_{\sigma_2} \min_{\sigma_1} \sigma_1^T A\sigma_2. \tag{2.9}$$

The *minimax theorem* states that every finite game has a value and guarantees the existence of a saddle-point equilibrium in mixed strategies.

Theorem 2.4 (Minimax Theorem) *In any matrix game A, the average security levels of the players in mixed strategies coincide, that is,*

$$\overline{V}_m(A) = \min_{\sigma_1} \max_{\sigma_2} \sigma_1^T A\sigma_2 = \max_{\sigma_2} \min_{\sigma_1} \sigma_1^T A\sigma_2 = \underline{V}_m(A). \tag{2.10}$$

2.2 Nonzero-Sum Games

The concept of two-person zero-sum matrix games can be extended to describe a strategic form N-player nonzero-sum game. Consider a set of players $\mathbb{PL} \triangleq \{1, 2, \cdots, N\}$. For each player $i \in \mathbb{PL}$, we can define a finite or infinite set of actions A_i. Player i can play a *pure strategy* $a_i \in A_i$ or a *mixed strategy* $\sigma_i \in \Sigma_i$, which is a probability distribution over the set A_i of pure strategies. Denote by a_{-i} the strategies of all players excluding player i. Then we can define the function $u_i : \prod_{i=1}^N A_i \to \mathbb{R}$ such that $u_i(a_i, a_{-i})$ gives the *payoff function* of each player $i \in \mathbb{PL}$. The *expected utility* of each player $i \in \mathbb{PL}$ is then denoted by $U_i(\sigma_i, \sigma_{-i}) \equiv \mathbb{E}_{\sigma_i, \sigma_{-i}} u_i(a_i, a_{-i})$.

Nash equilibrium (NE) is a natural solution concept for N-player nonzero-sum games. We call $(\sigma_i^*, \sigma_{-i}^*)$ a NE in mixed strategies if

$$U_i(\sigma_i^*, \sigma_{-i}^*) \geq U_i(\sigma_i, \sigma_{-i}^*), \tag{2.11}$$

for all admissible $\sigma_i \in \Sigma_i$ and for all $i \in \mathbb{PL}$. The interpretation of the equilibrium is that no player has an incentive to deviate unilaterally from the equilibrium point. When the utility function is well-behaved, we can also represent Eq. (2.11) using the best-response mapping $BR_i(\cdot) : \Pi_{j \neq i} \Sigma_i \to 2^{\Sigma_i}$, defined such that

$$BR_i(\sigma_{-i}) \equiv \arg\max_{\sigma_i \in \Sigma_i} U_i(\sigma_i, \sigma_{-i}).$$

Note that BR_i is a correspondence, i.e., a point-to-set mapping.

Now Eq. (2.11) can be rewritten as

$$\sigma_i^* \in BR_i(\sigma_{-i}^*), \quad \forall i \in \mathbb{PL}. \tag{2.12}$$

For finite N-person games, the existence of NE solutions in mixed strategies is guaranteed. This result is summarized in the following theorem.

Theorem 2.5 (Nash Equilibrium) *Every N-person static finite game in normal form admits a NE solution in mixed strategies [80].*

2.3 Extensive-Form Games

In this section, we introduce *extensive-form games* and behavioral strategies. These concepts are important in order to capture dynamic aspects of strategic interactions. We also describe a fundamental extensive-form game called a Stackelberg game [81].

2.3.1 Representation Using Graphs

Before we give a formal definition for extensive-form games, let us consider an example game called *matching pennies*. This game can be conceptualized by the extensive-form, imperfect-information games given in Fig. 2.1a, b. In these figures, the payoff pairs (e.g., (1,-1)) are assigned to Players P1 and P2, respectively, when P1 and P2 pick an action from H, T (*heads*, *tails*). P1 wants the actions of the players to *match*, while P2 desires a *mismatch*. The dashed oval in Fig. 2.1a indicates that P2 is not able to distinguish whether P1 has played H or T. Thus, his information is *imperfect*. The same is true of the dashed oval for P1 in Fig. 2.1b.

One can verify that the games in Fig. 2.1a, b are both equivalent to the bi-matrix, normal-form game expressed by Table 2.1.

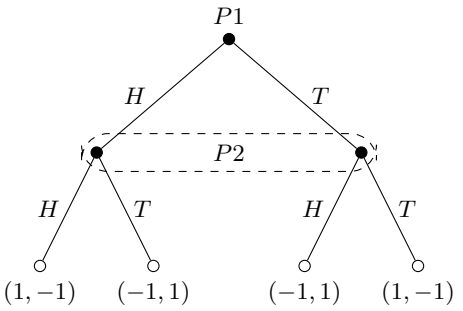
2.3.2 Information Sets

But to explicitly display the dynamic character of the game, one has to define the signal device to which the players have access, namely the *information set*. Here we formally give the definition of extensive-form games with perfect information, and then we define then information sets.

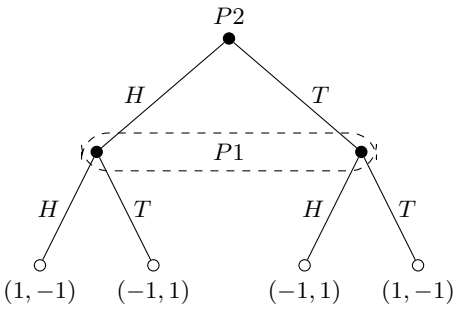
Definition 2.6 (*Extensive-form Games with Perfect Information and Chance Moves* [82]) A game in extensive form is a vector

$$G = \left(\mathbb{PL}, V, E, x^0, (V_i)_{i\in\mathbb{PL}\cup(0)}, (p_x)_{x\in V_0}, O, u\right), \quad \text{where}$$

Fig. 2.1 Matching pennies game in extensive form



(a) Form 1



(b) Form 2

Table 2.1 Matching pennies in normal form

		II	
		H	*T*
I	*H*	(1, −1)	(−1, 1)
	T	(−1, 1)	(1, −1)

- $\mathbb{PL}$ is the set of players.
- (V, E, x^0) is the game tree.[3]
- $(V_i)_{i \in \mathbb{PL} \bigcup \{0\}}$ is a partition of the set of vertices that are not leaves.
- For every vertex $x \in V_0$, p_x is a probability distribution over the edges emanating from x.
- O is the set of possible outcomes.
- u is a function mapping each leaf of the game tree to an outcome in O.

The notion of *information sets* is useful in order to model the players' strategies. Roughly speaking, an information set is a set of decision nodes such that

[3]Here, we use E, V, and V_i for $i \in \mathbb{PL} \bigcup \{0\}$ to represent sets, even though the symbols are not double-struck, in order to conform to accepted notation for graphs.

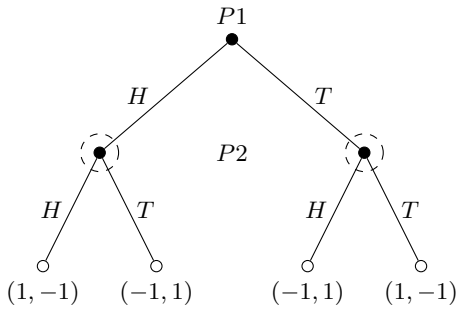


Fig. 2.2 Matching pennies with perfect information

- Every node in the set belongs to one player, and
- When the game reaches the information set, the player who acts cannot differentiate between the nodes within the information set. That is, if the information set contains more than one node, the player does not know which node in the set has been reached.

Definition 2.7 formally expresses this concept.

Definition 2.7 (*Information Sets* [82]) Consider the game

$$G = \left(\mathbb{PL}, V, E, x^0, (V_i)_{i\in\mathbb{PL}\cup\{0\}}, (p_x)_{x\in V_0}, O, u\right).$$

An information set of player i is a pair $(\eta_i, \mathbb{A}(\eta_i))$ such that

- $\eta_i = \left\{x_i^1, x_i^2, \ldots, x_i^m\right\}$ is a subset of V_i that satisfies the property that at each vertex in η_i, player i has the same number of actions l_i, i.e.,

$$\left|\mathbb{A}\left(x_i^j\right)\right| = l_i, \quad \forall j = 1, 2, \ldots, m.$$

- $\mathbb{A}(\eta_i)$ is a partition of the ml_i edges $\bigcup_{j=1}^m \mathbb{A}\left(x_i^j\right)$ to l_i disjoint sets, each of which contains one element from the sets $\left(\mathbb{A}\left(x_i^j\right)\right)_{j=1}^m$. We denote the elements of the partition by $a_i^1, a_i^2, \ldots, a_i^{l_i}$. The partition $\mathbb{A}(\eta_i)$ is called the action set of player i in the information set η_i.

The notion of information set was introduced by John von Neumann, motivated by studying the game of Poker. With some abuse of notation, in Fig. 2.1a the information set for $P2$ simply is $\eta_2 = \{H, T\}$, where H, T signifies the two nodes coupled together by the dashed oval.

Let us modify the extensive-form game depicted by Fig. 2.1a to that given by Fig. 2.2. Given the new information sets, $P2$ can distinguish between different acts of $P1$. $P2$ now has new choices for his strategies, which are

Table 2.2 Matching pennies with perfect information

		II			
		HH	HT	TH	TT
I	H	$(1,-1)$	$(-1,1)$	$(1,-1)$	$(-1,1)$
	T	$(-1,1)$	$(1,-1)$	$(1,-1)$	$(-1,1)$

$$\gamma_2^1 = \begin{cases} H & \text{if } a_1 = H \\ H & \text{if } a_1 = T \end{cases}$$

$$\gamma_2^2 = \begin{cases} T & \text{if } a_1 = H \\ T & \text{if } a_1 = T \end{cases}$$

$$\gamma_2^3 = \begin{cases} H & \text{if } a_1 = H \\ T & \text{if } a_1 = T \end{cases}$$

$$\gamma_2^4 = \begin{cases} T & \text{if } a_1 = H \\ H & \text{if } a_1 = T \end{cases}$$

Thus the normal form can be extended to Table 2.2, where HH, TT, HT, TH denote γ_2^1, γ_2^2, γ_2^3, γ_2^4, respectively. The modification of the information set extends $P2$'s strategy space to a set of four pure strategies. It can be verified that this creates a dominate strategy for $P2$. His situation has improved with more information; the claim that *knowledge is power*—although elsewhere false—is true in this example.

The strategy space given by Table 2.2 describes a *strategic-form game*. The actions of a strategic-form game are the strategies of the higher-level game. We can formulate mixed strategies for the strategic-form game using Definition 2.8.

Definition 2.8 (*Mixed Strategies*) Let $G = \left(\mathbb{PL}, (\Gamma_i)_{i\in N}, (u_i)_{i\in N}\right)$ be a strategic-form game in which the set of strategies of each player is finite. A mixed strategy of player i is a probability distribution over his set of strategies Γ_i. The space of mixed-strategies is given by

$$\Delta_i := \left\{\alpha_j : \Gamma_i \to [0,1] \; s.t. \sum_{\gamma_i^j \in \Gamma_i} \alpha_j(\gamma_i^j) = 1\right\}.$$

Note that these mixed strategies are for the strategic-form game given by Table 2.2. A function can also be formulated for the mixed-strategy utility, but we omit this here.

2.3.3 Behavioral Strategies

Behavioral strategies are an important class of strategies that arise naturally from the concept of information sets.

Definition 2.9 (*Behavioral Strategy*) Let I_i be the set of information sets of player i and let $\eta_i \in I_i$ be a generic information set. A behavioral strategy for player i is a function $\gamma_i(\cdot) : I_i \to \bigcup_{\eta_i \in I_i} \Delta(\mathbb{A}(\eta_i))$ such that $\gamma_i(\eta_i) \in \Delta(\mathbb{A}(I_i))$ for all $\eta_i \in I_i$, where I_i represents the collection of information sets, $\eta_i \in I_i$ is one of the information sets, $\mathbb{A}(\eta_i)$ is the action space at η_i, and $\Delta(\mathbb{A})$ represents the set of all feasible distributions over the action space $\mathbb{A}$. This function maps each of player i's information sets to a probability distribution over the set of possible actions at that information set.

Let us summarize Definitions 2.8 and 2.9. Given an extended-form game, players can randomly choose a pure strategy at the outset of the game; this type of randomization yields the concept of mixed strategies in an extensive-form game (Definition 2.8). Alternatively, at every one of their information sets players can randomly choose one of their available actions; this type of randomization describes behavioral strategies (Definition 2.9). Roughly speaking, a mixed strategy randomly chooses a deterministic path through the game tree, while a behavior strategy can be seen as a stochastic path.

2.3.4 Perfect Equilibria

A Single-Act Game with Dynamic Information
Consider the extensive-form game given by Fig. 2.3, in which both players are minimizers. The game can be solved by backward induction. We start by finding the optimal (behavioral) strategy for P2, who moves second. His information sets

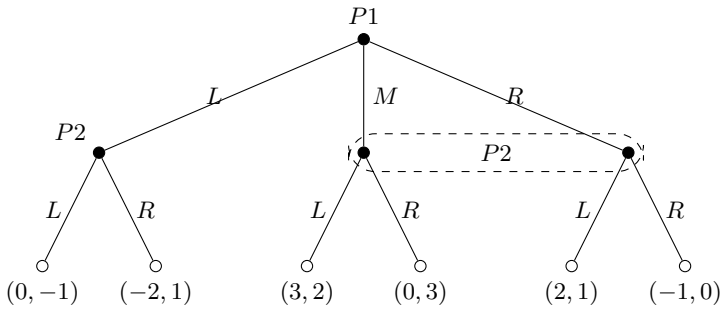


Fig. 2.3 Example two-player extended-form game

Table 2.3 RHS game

		II	
		L	R
I	M	(3, 2)	(0, 3)
	R	(2, 1)	$(-1, 0)^{\star}$

are given by $I_2 = \{\{L\}, \{M, R\}\}$. If P1 plays L, it is obvious that P2's best move is L, so that he secures a utility of -1 rather than 1. We have $u_1 = 0$, $u_2 = -1$.

What should P2 play at the information set $\{M, R\}$? In fact, if we remove the sub-tree that results from $a_1 = L$, we are left with a tree similar to Fig. 2.1a, b. This sub-game should be solved using NE. Table 2.3 gives the normal form of the sub-game induced by $\eta_2 = \{M, R\}$. The NE is $\{R, R\}$, which yields the payoff pair $u_1 = -1$, $u_2 = 0$. Hence, we can summarize the behavioral strategy of P2 by

$$\gamma_2^*(\eta_2) = \begin{cases} L & \text{if } \eta_2 = \{L\} \\ R & \text{otherwise} \end{cases}.$$

Moving to the top-level node, P1 should choose R rather than L in order to secure a utility of -1 rather than 0. Hence, $\gamma_1^*(\eta_1) = R$. The complete behavioral-strategy equilibrium is given by γ_1^*, γ_2^*.

Is this the same as the equilibrium that results from the strategic-form game? The strategy spaces given by the strategic-form game are $\Gamma_1 = \{L, M, R\}$ and $\Gamma_2 = \{LL, LR, RL, RR\}$, where these last four strategies signify

$$\gamma_2^1 = \begin{cases} L & \text{if } u_1 = L \\ L & \text{if } u_1 = M, R \end{cases},$$

$$\gamma_2^2 = \begin{cases} L & \text{if } u_1 = L \\ R & \text{if } u_1 = M, R \end{cases},$$

$$\gamma_2^3 = \begin{cases} R & \text{if } u_1 = L \\ R & \text{if } u_1 = M, R \end{cases},$$

$$\gamma_2^4 = \begin{cases} R & \text{if } u_1 = L \\ L & \text{if } u_1 = M, R \end{cases}.$$

The strategic form of the game can then be represented by Table 2.4.

We observe that there are three NE according to the strategic form, while behavioral strategies admit only one NE. In fact, it can be remarked that the strategic-form solution is generally not equivalent to the solution in behavioral strategies. Behav-

Table 2.4 Game from Fig. 2.3 in strategy space

		II			
		LL	LR	RR	RL
I	L	$(0, -1)^\star$	$(0, -1)$	$(-2, 1)^\star$	$(-2, 1)$
	M	$(3, 2)$	$(0, 3)$	$(0, 3)$	$(3, 2)$
	R	$(2, 1)$	$(-1, 0)^\star$	$(-1, 0)$	$(2, 1)$

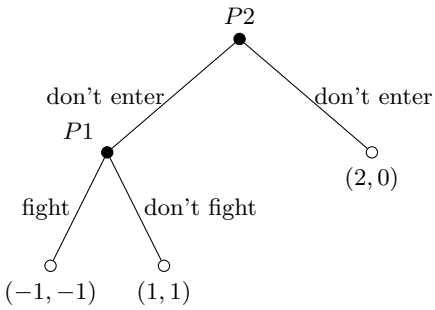


Fig. 2.4 In this game, not all NE are behavioral-strategy equilibria

Table 2.5 Market entry game in matrix form

		II	
		E	D
I	F	$(-1, -1)$	$(2, 0)^\star$
	A	$(1, 1)^\star$	$(2, 0)$

ioral strategies select equilibria from the set of strategic-form equilibria according to two criteria:

- The equilibria must be credible, which means that they represent NE of every sub-game of the original dynamic game.
- The equilibria satisfy the trembling-hand property, which means that they are robust to small perturbations and uncertainties.

Market Entry Game

As another example, consider a situation in which firm $P2$ decides whether to enter the market which $P1$ already occupies. Both players are maximizers.

The strategy spaces are given by $\gamma_1 \in \{E, D\}$ (enter, don't enter) and $\gamma_2 \in \{F, A\}$ (fight, don't fight). The open-loop form is given by Table 2.5.

While (A, E) and (F, D) are both NE, (F, D) is *not credible*. The reason is that if P2 were to deviate and play E, then $a_1 = D$ would not be an optimal response. Hence, (F, D) is not an equilibrium in behavioral strategies. (A, E) is selected as the behavioral-strategy equilibrium.

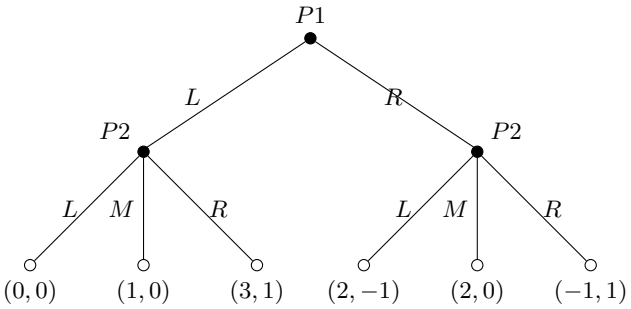


Fig. 2.5 An example of a Stackelberg game. Both players are minimizers

2.4 Stackelberg Games

NE provides a reasonable noncooperative equilibrium solution for nonzero-sum games in which the roles of the players are symmetric, i.e., in which no single player dominates the decision process. In other types of noncooperative decision problems, however, one of the players has the ability to enforce his strategy on the other player(s), and such decision problems motivate the introduction of a hierarchical equilibrium concept.

2.4.1 Stackelberg Game with Non-unique Best Responses

In fact, we have already seen two Stackelberg games: those given by Figs. 2.2 and 2.4. In these games, the optimal response of the follower was unique for each possible action of the leader. But what happens if the followers have multiple optimal strategies?

Consider the Stackelberg game depicted in Fig. 2.5, in which both players are minimizers. An equilibrium does in fact exist if we stipulate that the leader prefers to secure his possible losses against any choices of the follower that fall within his set of optimal responses. Under such a stipulation, P1's secured cost level corresponding to his strategy L is 1, and the cost corresponding to R is 2.

More formally, consider two players $P1$ and $P2$ who maximize their payoff functions $u_1(a_1, a_2)$ and $u_2(a_1, a_2)$, where $a_1 \in \mathbb{A}_1$ and $a_2 \in \mathbb{A}_2$. In a Stackelberg equilibrium, P2 chooses his action according to Definition 2.10.

Definition 2.10 (*Best Response*) The optimal response set[4] of $P2$ to strategy $a_1 \in \mathbb{A}_1$ of $P1$ is

$$R_2(a_1) = \{\xi \in A_2 : U_2(a_1, \xi) \geqslant U_2(a_1, a_2)\}. \tag{2.13}$$

R_2 is nonempty if the game is finite.

Given this response, P1 chooses his action according to Definition 2.11.

Definition 2.11 (*Optimal Leader Action*) Action $a_1^* \in \mathbb{A}_1$ satisfies a Stackelberg equilibrium for the leader if

$$\min_{a_2 \in R_2(a_1)} u_1(a_1^*, a_2) = \max_{a_1 \in A_1} \min_{a_2 \in R_2(a_1)} u_1(a_1, a_2) \equiv u_1^*. \tag{2.14}$$

where u_1^* is the equilibrium cost of the leader.

The Stackelberg strategy for the leader does not necessarily have to be unique. But non-uniqueness of the equilibrium strategy is not problematic, since the Stackelberg *cost* for the leader is unique.

If $R_2(a_1)$ is a singleton set for each $a_1 \in \mathbb{A}_1$, then there exists a mapping $T_2 : \mathbb{A}_1 \to \mathbb{A}_2$ such that $a_2 \in R_2(a_1)$ implies that $a_2 = T_2(a_1)$. This corresponds to the case in which the optimal response of the follower (T_2) is unique for every strategy of the leader, and it leads to the following simplified form of Eq. (2.14):

$$u_1(a_1^*, T_2(a_1^*)) = \max_{a_1 \in \mathbb{A}_1} u_1(a_1, T_2(a_1)) \equiv u_1^*.$$

Thus u_1^* becomes a cost value that the leader can truly attain.

Definition 2.12 (*Stackelberg Outcome*) Let $a_1^* \in \mathbb{A}_1$ be a Stackelberg strategy for leader $P1$. Then any element $a_2^* \in R_2(a_1^*)$ is an optimal strategy for the follower. The action profile $\{a_1^*, a_2^*\}$ is a Stackelberg equilibrium, and $u_1(a_1^*, a_2^*)$ and $u_2(a_1^*, a_2^*)$ are the corresponding Stackelberg outcomes.

Note that in Definition 2.12, $u_1(a_1^*, a_2^*)$ could be greater, (i.e., the cost level could be lower) than the Stackelberg cost u_1^*, a feature which has already been observed within the context of the example. If $R_2(a_1^*)$ is a singleton, however, then these two cost levels coincide.

2.5 Notes

Sections 2.1–2.4 provide a brief introduction to zero-sum games, nonzero-sum games, extensive form games and their solution concepts. The exposition of this chapter follows [28]. Interested readers can refer to Chaps. 2 and 3 of the textbook [28].

[4] Here also R_2 represents a set even though it is not double-struck.

The relationship between mixed strategies and behavioral strategies (Sect. 2.3) has been studied by Harold W. Kuhn in [83]. Interested readers can find a detailed exposition of the material in Chap. 6 of [82] and Chap. 3 of [28].

Chapter 3
Introduction to Incomplete Information

In this chapter, we introduce games with incomplete information. The chapter first gives an overview of Bayesian games and then presents signaling games as an example of two-stage games of incomplete information. Signaling games will be used in several chapters within the book (Chaps. 4, 6–8).

3.1 Bayesian Games

Bayesian games are one type of *incomplete information* games. A Bayesian game can be expressed by the tuple $\langle \mathbb{PL}, (\mathbb{A}_i)_{i\in\mathbb{PL}}, (u_i)_{i\in\mathbb{PL}} \rangle$, where $\mathbb{PL}$ is the set of the players, $\mathbb{A}_i$ is the action set of player i, and u_i is the utility function of player i. Incomplete information means that player i knows certain private information, (e.g., the utility function) that belongs to himself, but does not know the other players' private information.

Harsanyi's framework captures all the uncertainties and the hierarchy of knowledge into one single variable, which is called the *type* or *attribute* parameter. Each player is characterized by his own type. Player i knows his own type, but he does not know other players' types, nor do other players know player i's type. We do assume, however, that the distribution of types of each player is commonly known.

3.1.1 N-Person Bayesian Nash Equilibrium Problem

Consider the following scenario. Let $\mathbb{PL} = \{1, 2, \ldots, N\}$ be a set of players. Each player is associated with a type/attribute (private information) $\theta_i \in \Theta_i$. The realization θ_i is private but the distribution of the type and the associated support is commonly known. The type θ_i is drawn from the distribution p_i, $i = 1, \ldots, N$. In general, p_i may depend only on θ_i or may depend also on the types of other players, denoted by θ_{-i}. Each player i has an action space $\mathbb{A}_i$, which represents the set of all

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_3

his possible actions. The *strategy* for player i is a mapping $\zeta_i : \Theta_i \mapsto \mathbb{A}_i$, such that

$$a_i = \zeta_i(\theta_i) \quad i = 1, \ldots, N.$$

It is clear that player i's strategy is a function of the attribute known to the player himself. Player i can determine action a_i from the mapping, while he can only anticipate other players' actions from their mappings $\zeta_{-i} := \{\zeta_j, j \neq i, j \in \mathbb{PL}\}$, without knowing their types. The payoff for player i is

$$u_i(a_1, \ldots, a_n; \theta_i, \theta_{-i}) = \tilde{u}_i\left(\zeta_1(\theta_1), \ldots, \zeta_N(\theta_N); \theta_i, \theta_{-i}\right) \quad i = 1, \ldots, N.$$

Here, $u_i(\cdot)$ is defined on the action space while $\tilde{u}_i(\cdot)$ is defined on the strategy space. The equilibrium concept for Bayesian games is similar to that of Nash equilibrium.

Definition 3.1 (*Pure Strategy Bayesian Nash Equilibrium*) We call $\{\zeta_i^*\}_{i=1}^N$ a (pure strategy) Bayesian Nash equilibrium if, $\forall i \in \mathbb{PL}$, $a_i^* = \zeta_i^*(\theta_i)$ and, $\forall \theta_i \in \Theta_i$, $i \in \mathbb{PL}$,

$$\max_{a_i \in \mathbb{A}_i} \mathbb{E}_{p_{-i}|\theta_i}\Big[u_i\big(\zeta_1(\theta_1), \ldots, \zeta_{i-1}(\theta_{i-1}), a_i, \zeta_{i+1}(\theta_{i+1}), \ldots, \zeta_N(\theta_N); \theta_i, \theta_{-i}\big)|\theta_i\Big].$$

Note that the uncertainties from the type variables are captured in the utility function. The expectation is taken over the uncertainties of the types of other players given player i's own type. The Bayesian Nash equilibrium $\{\zeta_i^*\}_{i=1}^N$ is a strategy profile in which no players have incentives to deviate away from their behaviors by unilaterally changing their strategies.

3.1.2 Two-Bidder Auction as an Example

In order to illustrate games of incomplete information, let us examine a two-bidder auction problem. Consider a two-player auction in which each buyer has a valuation v_i, $i = 1, 2$, known only to the buyer himself. The valuation v_i is a realization of the random variable $\mathcal{V}_i$ with support $[0, v_{i,max}]$ and associated probability density function[1] p_i, $i = 1, 2$.

Each bidder chooses a bid b_i, based on his type v_i. All bids are submitted simultaneously. The bidders do not know other players' bids. The bidder with the highest bid wins the good. The winner has to pay his bid, i.e., the highest bid.[2] The payoff function for player i, $i = 1, 2$, is then given by

[1] Note that (v_1, v_2) can be correlated, so there might be a joint distribution function. Here, for simplicity, we assume that they are independent.

[2] Other payment rules such as the second price rule or the third price rule can also be used.

$$u_i(b_1, b_2, v_1, v_2) = \begin{cases} v_i - b_i & b_i > b_{-i} \\ 0 & b_i < b_{-i} \\ \frac{1}{2}(v_i - b_i) & b_i = b_{-i} \end{cases}.$$

The last row of u_i means that each bidder has a one-half chance to win the good if the players bid the same quantity.

An *ex ante* solution is formed from the strategies that consider all the contingencies before nature plays, i.e., before the bidder is informed of his own type. Mathematically, the ex ante solution is written as

$$b_i = \zeta_i(v_i), \quad i = 1, 2,$$

where $\zeta_i : [0, v_{i,max}] \to \mathbb{R}_+$ is called the *bidding function*. Since each bidder does not know his own type at the ex ante stage, he needs to prepare for all possible contingencies. Hence, to find an ex ante solution is equivalent to finding a bidding function ζ_i.

An *ex post* solution is formed from the strategy chosen after nature plays, i.e., after the bidder is informed of his type. Mathematically, the ex post solution is a value (rather than a function) $\zeta_i(v_i)$, $i = 1, 2$. For player 1, the ex ante solution is

$$\max_{\zeta_1(\cdot)} \mathbb{E}_{p_2} \left[u_1(\zeta_1(v_1), \zeta_2(v_2), v_1, v_2) \right]. \tag{3.1}$$

This optimization problem is difficult, because it involves finding an optimizing function. But the problem is equivalent to solving the following problem:

$$\max_{b_1 \in \mathbb{R}} \mathbb{E}_{p_2} \left[u_1(b_1, \zeta_2(v_2), v_1, v_2) | v_1 \right], \tag{3.2}$$

which gives the ex post solution. We can solve for b_1^* for every v_1, and then the optimal bidding strategy can be constructed as a function of v_1. This function is $\zeta_1^*(\cdot)$ such that $b_1^* = \zeta_1^*(v_1)$. It solves Eq. (3.1). The ex ante solution is consistent with the ex post solution.

For bidder 2, in a similar way, we can find $b_2^* = \zeta_2^*(v_2)$ that solves

$$\max_{b_2 \in \mathbb{R}} \mathbb{E}_{p_1} \left[u_2(\zeta_1(v_1), b_2, v_1, v_2) | v_2 \right]. \tag{3.3}$$

Solving Eqs. (3.2) and (3.3) gives a pair of policies (ζ_1^*, ζ_2^*) which simultaneously satisfy the following equations:

$$\zeta_1^* \in \arg\max_{\zeta_1(\cdot)} \mathbb{E}_{p_2} \left[u_1(\zeta_1(v_1), \zeta_2(v_2), v_1, v_2) \right],$$
$$\zeta_2^* \in \arg\max_{\zeta_2(\cdot)} \mathbb{E}_{p_1} \left[u_2(\zeta_1(v_1), \zeta_2(v_2), v_1, v_2) \right].$$

The pair (ζ_1^*, ζ_2^*) constitutes a Bayesian Nash equilibrium for the two-person auction game. The following theorems are extensions of the minimax theorem (Theorem 2.4) and Nash's theorem (Theorem 2.5) to Bayesian games. They provide existence results for finite and infinite Bayesian games.

Proposition 1 (Existence for Finite Games) *Consider a finite, incomplete-information (Bayesian) game. Then a mixed strategy Bayesian Nash equilibrium exists.*

Proposition 2 (Existence for Infinite Games) *Consider a Bayesian game with continuous strategy spaces and continuous types. If strategy spaces and type spaces are compact, and payoff functions are continuous and concave in players' own strategies, then a pure strategy Bayesian Nash equilibrium exists.*

3.2 Signaling Games

Signaling games are a special type of dynamic games of incomplete information. Signaling games involve two players. The first player is the sender, and the second player is the receiver. The sender (S) has private information modeled by type θ, and moves first by choosing an action or message m based on the type. The message is sent to the receiver (R) (perfectly, without distortion). The receiver chooses an action a in order to respond to the message that he observes.

3.2.1 Signaling-Game Model

The problem setup is depicted in Fig. 3.1 and described as follows.

- Nature draws a type θ for the sender from a set of feasible types Θ according to a probability distribution $p(\theta)$, where $p(\theta) > 0$ for every θ, and $\sum_{\theta \in \Theta} p(\theta) = 1$.
- The sender observes θ and then chooses a message m from a set of feasible messages $\mathbb{M}$ using the mixed strategy $\sigma^S(m|\theta)$. Here, $\sigma^S(m|\theta)$ is the probability of choosing $m \in \mathbb{M}$ when the sender is of type θ.
- The receiver observes m, and then chooses an action a from a set of feasible actions $\mathbb{A}$ using the mixed strategy $\sigma^R(a|m)$. Here, $\sigma^R(a|m)$ yields a probability of choosing $a \in \mathbb{A}$ if the receiver observes m.
- Payoffs are given by $u^S(\theta, m, a)$ and $u^R(\theta, m, a)$.

As an example, consider the two-type, two-message, two-action signaling game shown in Fig. 3.2. Here, $\Theta = \{\theta_1, \theta_2\}$, $M = \{L, R\}$, $A = \{T, D\}$ and $p(\theta_1) = p(\theta_2) = 1/2$. The dashed lines represent information sets. (See Definition 2.7 in Chap. 2.)

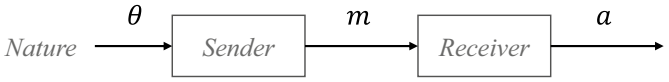


Fig. 3.1 In the basic signaling game framework, a sender transmits a message m to a receiver, who chooses an action a. The sender has private information θ unknown to the receiver. This private information can be considered as a chance move by "nature"

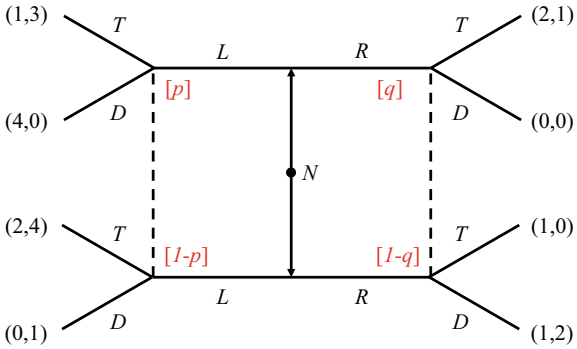


Fig. 3.2 Signaling games with binary type, message, and action spaces can be depicted in extensive-form diagrams in which play initiates at the central node N (nature)

In this example, the sender has 4 pure strategies:

- (L, L): Play L if nature draws θ_1, and play L if nature draws θ_2.
- (L, R): Play L if nature draws θ_1, and play R if nature draws θ_2.
- (R, R): Play R if nature draws θ_1, and play R if nature draws θ_2.
- (R, L): Play R if nature draws θ_1, and play L if nature draws θ_2.

The receiver also has 4 pure strategies:

- (T, T): Play T if the sender chooses L, and play T if the sender chooses R.
- (T, D): Play T if the sender chooses L, and play D if the sender chooses R.
- (D, D): Play D if the sender chooses L, and play D if the sender chooses R.
- (D, T): Play D if the sender chooses L, and play T if the sender chooses R.

Sender strategies such as (L, L) and (R, R) are called *pooling strategies*. The sender chooses the same message for both types. Sender strategies such as (L, R) and (R, L) are called *separating strategies*. The sender chooses different messages for each type. *Partially separating strategies* are also feasible, and are explored in Chap. 6.

3.2.2 Signaling-Game Equilibrium Concept

A common solution concept for signaling games is perfect Bayesian Nash equilibrium (PBNE). It requires the following conditions to be held consistently.

Condition 1 (Belief Formation) *The receiver forms a belief about the type after observing the message $m \in \mathbb{M}$ sent by the sender. This belief is denoted by the conditional distribution $\mu(\theta|m)$, where $\mu(\theta|m) \geq 0$ for each $\theta \in \Theta$, and for which*

$$\sum_{\theta \in \Theta} \mu(\theta|m) = 1, \quad m \in \mathbb{M}.$$

Condition 2 (Receiver's Problem) *For every message m that is received, the receiver maximizes his expected utility given his belief $\mu(\theta|m)$. That is,*

$$\max_{\sigma^R} \sum_{a \in \mathbb{A}} \sum_{\theta \in \Theta} \mu(\theta|m) \sigma^R(a|m) u^R(\theta, m, a).$$

As a result, the receiver's strategy is given by the mapping σ^{R*}.

Condition 3 (Sender's Problem) *For every type $\theta \in \Theta$, the sender's message maximizes the following utility, anticipating the receiver's strategy σ^{R*}:*

$$\max_{\sigma^S} \sum_{m \in \mathbb{M}} \sum_{a \in \mathbb{A}} \sigma^S(m|\theta) \sigma^{R*}(a|m) u_S(\theta, m, a).$$

As a result, the receiver's strategy is given by the mapping σ^{S*}.

This condition is in the same spirit as in the definition of Stackelberg equilibrium.

Condition 4 (Bayes Rule Consistency) *The strategy $\sigma^{S*}(m|\theta)$ is used to update the receiver's belief $\mu(\theta|m)$ according to Bayes' rule:*

$$\mu(\theta|m) = \frac{p(\theta)\sigma^{S*}(m|\theta)}{\sum_{\theta' \in \Theta} p(\theta')\sigma^{S*}(m|\theta')}, \quad \text{if} \sum_{\theta' \in \Theta} p(\theta')\sigma^{S*}(m|\theta') > 0,$$

$$\mu(\theta|m) = \text{any probability distribution}, \quad \text{if} \sum_{\theta' \in \Theta} p(\theta')\sigma^{S*}(m|\theta') = 0.$$

Note that these four conditions are coupled. The PBNE requires the conditions to be simultaneously satisfied.

Definition 1 (*Perfect Bayesian Nash Equilibrium*) A pair of strategies σ^{S*}, σ^{R*} and belief μ give a pure-strategy PBNE of the signaling game if they satisfy the four Conditions 1–4.

One way to find PBNE is to fix μ and solve for σ^{S*} and σ^{R*} (an equivalent process to finding Stackelberg equilibrium), and then find a μ that is consistent with the obtained σ^{S*} and σ^{R*}. This approach may require several iterations between μ and σ^{S*}, σ^{R*} for complicated games.

For the example game shown in Fig. 3.2, we can find two pure-strategy PBNE. One is (L, R), (T, D), supported by the belief $\mu(\theta_1|L) = 1$, $\mu(\theta_1|R) = 0$, and the other is (L, L), (T, D), supported by the belief $\mu(\theta_1|L) = 1/2$ and any $\mu(\theta_1|R) \leq 2/3$.

3.3 Notes

The theory of games of incomplete information (Sect. 3.1) dates back to Harsanyi's three-part work in the 1950s. Readers can refer to Harsanyi's original work [84–86]. Myerson has also written a short commentary paper [87] on Harsarny's work.

An equivalent model for Harsanyi's games of incomplete information is given by Aumann. The Aumann model captures not only the knowledge of the players regarding the payoff-relevant parameters, but also the whole hierarchy of knowledge of each player. Interested readers can read Chaps. 9–11 of the textbook [82] by Maschler et al. For more information about auctions, interested readers can refer to [88].

There is a rich literature on signaling games (Sect. 3.2) and their applications. The example presented in this chapter follows Chap. 4 of [89]. Interested readers can refer to the applications of signaling games in job markets, corporate investment, and monetary policy found in the book. The extension of two-stage signaling games to multistage games of incomplete information is discussed in Chap. 8 of [90]. Detailed discussions of trembling hand properties are provided in Chap. 8.4 of the same book.

Part II
Defensive Deception

Chapter 4
A Taxonomy of Defensive Deception

The game theory described in Chaps. 2 and 3 offers versatile possibilities for quantifying cyber deception. Yet specific game-theoretic models must be carefully chosen in order to model different types of deception. In this chapter, therefore, we propose a taxonomy that classifies deception into six categories. We then propose game-theoretic models that fittingly describe each category. Our analysis is based on both theoretical considerations and an empirical study of 25 recent articles in game theory for defensive cyber deception. The taxonomy provides a systematic foundation for understanding the three types of defensive deception that we study in Chaps. 5–7.

4.1 Introduction

Any scientific approach to cyber deception should start by defining fundamental terms. The term "deception," however, has been employed broadly and with a variety of meanings. Consider the following definition [91]:

> To deceive = to intentionally cause another person to acquire or continue to have a false belief, or to be prevented from acquiring or cease to have a true belief.

This definition is broad, which matches the diversity of applications of defensive deception to cybersecurity and privacy. On the other hand, the breadth of the term deception limits its depth.

Distinctions among types of deception are already present in the literature to some extent. Authors use specific terms such as moving target defense [49, 92], obfuscation [18, 93, 94], and honey-x (honeypots, honeynets, honeytokens, etc.) [95–100] to refer to different kinds of deception. Nevertheless, the meaning of these terms is not completely clear. For instance, what is the difference between moving target defense and obfuscation? How should game-theoretic models used to study honeypots differ from those used to model privacy? These types of questions motivate the development of a game-theoretic taxonomy of cyber deception.

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_4

We can identify two stages in constructing such a taxonomy: a *descriptive stage* and a *prescriptive stage*. The descriptive stage consists of examining existing deception literature, clustering together similar types of deception, and listing the models used to study each type. The prescriptive stage involves formalizing the boundaries between different types of deception and selecting the game-theoretic models that seem best suited to study each type.

Section 4.2 is the descriptive stage of constructing our taxonomy. Besides listing deception techniques and game-theoretic models found in the literature, it can also serve as a point of departure for readers interested in related work. From a prescriptive approach, Sects. 4.3 and 4.4 select the definitions and models that we think are most promising. Types of deception are delineated by *specific differences* that correspond to the game-theoretic concepts of private information, players, actions, and time-horizon. These specific differences also determine models for each deception type. The result is a game-theoretic taxonomy of six types of defensive deception.

Chapters 5–7 present detailed models of three of these six types. We choose the three types simply because they are the ones that the authors of this book have studied in the most depth. Chapters 5–7 use the game-theoretic models that our prescriptive analysis in the present chapter identifies as most promising. Since these are not the only possible models, Chaps. 5–7 admit of some subjectivity. We refer readers who are interested in alternative approaches to the individual articles that we mention in Table 4.1.

4.2 Description of Existing Literature

Table 4.1 provides basic information for these 25 articles. They are taken from journal papers or conference proceedings published between 2009 and 2019. We have only included articles that (1) study cybersecurity or privacy, (2) use defensive deception, and (3) employ game theory. This excludes, for example, studies of defensive deception that do not use game theory. It also excludes game theory papers that study offensive rather than defensive deception. Most of these 25 papers were selected from the first 100 Google Scholar results for a search using the three aforementioned qualifiers.

4.2.1 Broad Clusters of Deception Techniques

The "keywords" column of Table 4.1 shows words from the title of each paper that could potentially represent the name of a type of deception. The keywords that can be found multiple times are *obfuscation*, *moving target defense*, and *honeypot*. Examining the papers with these keywords, we find several characteristics.

Table 4.1 Keywords and models for selected literature (c.f., [101])

Authors and year	Keywords	Application	Model
Chessa et al. 2015	–	Inform. privacy	Nash
Shokri 2015	Obfuscation	Inform. privacy	Stackelberg
Alvim et al. 2017	–	Inform. privacy	Nash
Theodorakopoulos et al. 2014	–	Location privacy	Bayesian Stackelberg
Rass et al. 2017	–	General security	Nash
Sengupta et al. 2018	Moving target defense	Network security	Stackelberg
Zhu and Başar 2013	Moving target defense	Network security	Nash
Feng et al. 2017	Moving target defense	General security	Stackelberg
Clark et al. 2012	–	Network security	Stackelberg
Zhu et al. 2012	–	Network security	Stackelberg
Pawlick and Zhu 2016	Obfuscation	Inform. privacy	Stackelberg
Pawlick and Zhu 2017a	Obfuscation	Inform. privacy	Mean-field + Stackelberg
Zhang et al. 2010	–	Anonymity	Best response in a repeated game
Freudiger et al. 2009	–	Location privacy	Bayesian nash
Lu et al. 2012	Pseudonym	Location privacy	Nash
Carroll and Grosu 2011	–	Network security	Signaling
Çeker et al. 2016	–	Denial of service	Signaling
Mohammadi et al. 2016	Avatars	Social networks	Signaling
Pawlick et al. 2019	–	Network security	Signaling
Píbil et al. 2012	Honeypot	Network security	Bayesian nash
Kiekintveld et al. 2015	Honeypots	Network security	Bayesian nash
Zhuang et al. 2010	Secrecy	General security	Multi-period signaling game
Durkota et al. 2015	Hardening	Network security	Stackelberg
Pawlick et al. 2019	Attacker engagement	Network security	Stackelberg + MDP
Horák et al. 2017	–	Network security	Partially-observable stochastic game

- In papers that study *obfuscation*, defenders waste effort and resources of attackers by directing them to decoy targets rather than real assets, or they protect privacy by revealing useless information aside real information. Recent practical implementations include TrackMeNot [37], which uses randomly-generated searches to protect those who use search engines from data profiling, and ConcealGAN

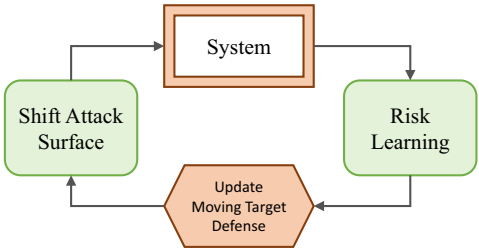


Fig. 4.1 Example of moving target defense from [49]. An attacker and defender play a series of zero-sum Nash games in which the defender chooses the arrangement of systems with various vulnerabilities, and the attacker selects an attack path that depends for its success on the arrangement of the vulnerabilities. Since these decisions are simultaneous, the defender strategy creates a moving target that limits the effectiveness of the attacker. The defender also updates the configuration of the system each round based on information learned about the attack risk that the system faces

[102], which hides text messages from censorship using meaningless cover texts. Chapter 5 will present one possible model for obfuscation.[1]

- In articles on *moving target defense,* defenders limit the effectiveness of attacker reconnaissance through techniques such as randomization and reconfiguration of networks, assets, and defense tools. Figure 4.1 gives one example of moving target defense.
- Techniques that use honeypots draw attackers towards specific target systems by disguising these machines as valuable network assets. Honeypots are used by commercial enterprises and government agencies to provide a high quality of alerts with low demand in human and computing resources. Honeyfiles are a similar technology used to manage compliances and protect enterprises from insider threats [103, 104]. Honeynets, such as those developed by the Honeynet Project, have been used to study attacks such as the Conficker botnet [105]. We use the term *honey-x* to refer to technologies such as honeypots, honeyfiles, and honeynets. Chapter 6 studies this type of deception.

Examining the remaining papers suggests three other clusters of deception techniques. We choose names for these types of deception from terms that are found within the bodies of the reviewed articles. All remaining articles can be considered as one of these three types.

- To protect privacy, defenders can add noise to published data. Let us call this type of deception *perturbation*. Apple implements one type of perturbation called "differential privacy" [106]. Google also recently open-sourced their own tool for this type of perturbation [107]. Figure 4.2 gives an example of perturbation studied using game theory.

[1] TrackMeNot, ConcealGAN, and other practical deception implementations mentioned in this section do not use game theory, so they are not included the 25 research works mentioned in Table 4.1.

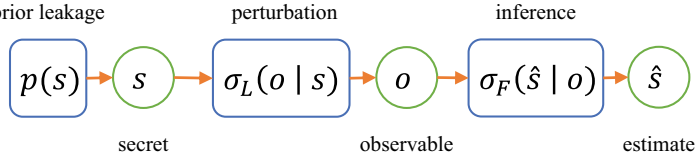


Fig. 4.2 Example of perturbation from [93]. Given a secret s, a user releases a perturbed observable o to an untrusted actor according to the channel probability $\sigma_L(o \mid s)$. The adversary uses inference $\sigma_F(\hat{s} \mid o)$ to obtain estimate $\hat{s}$. Perturbation limits the private information that the adversary learns about the user. The interaction is modeled using a non-zero-sum Stackelberg game

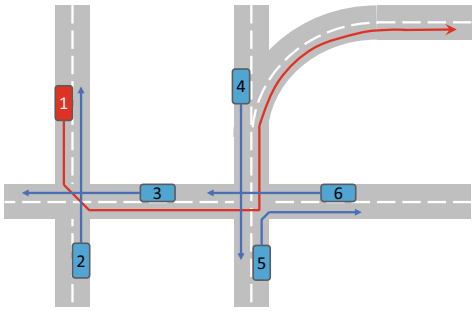


Fig. 4.3 Freudiger et al. [108] study pseudonym swapping for location privacy in vehicle ad hoc networks. As Vehicle 1 turns left at its first intersection, it can swap pseudonyms with Vehicle 2 and Vehicle 3. At its second intersection, it can swap with Vehicles 4, 5, and 6. As pseudonyms are mixed throughout the network, linkability of pseudonyms and vehicles decreases. Vehicles that have recently swapped have little incentive to swap again. This gives rise to interactions modeled by N-player, Bayesian Nash games

- Deception can use exchange systems such as mix networks and mix zones to prevent linkability. We call this type of deception *mixing*. See Fig. 4.3 for an example of mixing in vehicle networks.
- Defenders can use feedback to dynamically deceive attackers over an extended period of time. Let us call this *attacker engagement*. Chapter 7 will present a model for this type of deception.

4.2.2 Broad Modeling Trends

Which game-theoretic models do authors use to study these types of deception? Obviously, they do not agree on a one-to-one mapping between types of deception and games. This is to be expected, since (1) different games capture different aspects of the same interaction, (2) modeling approaches are still evolving in this nascent field, and (3) the cybersecurity landscape itself is evolving.

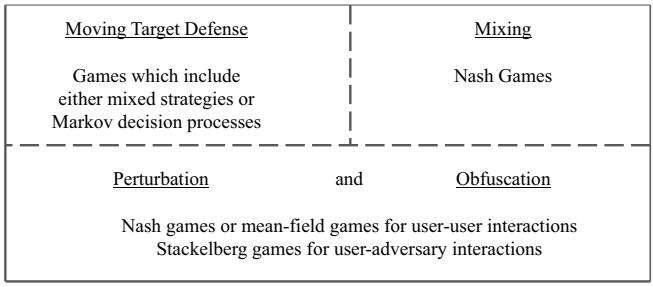


Fig. 4.4 Overview of models used to study moving target defense, mixing, perturbation, and obfuscation in the papers listed in Table 4.1

Nevertheless, some trends are evident. These are depicted in Figs. 4.4 and 4.5. Figure 4.4 shows that all of the papers on mixing use Nash games or related concepts. This is logical, because each user in a mix network or a mix zone simultaneously decides whether to participate. A second trend spans both perturbation and obfuscation. Papers in these areas that study user-user interactions are modeled using Nash games, and papers which study user-adversary interactions are generally modeled using Stackelberg games. Moving target defense tends to be modeled in two different ways. One way is to use mixed strategies to model randomness in defensive configurations. The second way is to use a Markov decision process (MDP) to explicitly model the temporal component of changing defenses.

Figure 4.5 shows that honey-x is modeled using two different approaches. One approach uses signaling games, in order to emphasize the formation of the attacker's belief about whether systems are normal systems or honeypots. The other approach uses Bayesian Nash games. This approach follows along the lines of resource allocation problems, and obtains an overall network configuration that is optimal for the defender. The right-hand side of the figure shows three approaches to attacker engagement. Zhuang et al. [109] use a multiple-period game which has a state that reflects information from past periods. The solution is obtained using dynamic programming. The authors of [110] represent a dynamic network attack using a Markov decision process. The attacker deploys the whole Markov decision process as a Stackelberg follower, and the defender places honeypots as a Stackelberg leader. Finally, [111] uses a one-sided partially-observable Markov decision process. This model is perhaps the closest of the three to a general *competitive Markov decision process*, in which a defender and an attacker choose dynamic policies to optimize long-term utility functions [112].

In summary, existing literature tends to cluster around six principle types of defensive deception, and each type tends to be modeled by one of several game-theoretic approaches. But the exact boundaries between deception types are not well defined. Nor do the clusters reveal any hierarchical relationships that may exist between the deception types. This imprecise language complicates the effort to accurately model deception with game theory. There is a need for "the construction of a common lan-

Honey-X	Attacker Engagement
Emphasis on attacker belief: Signaling games	Multiple-period games
	Interaction between games and MDPs
Emphasis on defender defense allocation: Bayesian Nash games	One-sided stochastic games

Fig. 4.5 Overview of models for honey-x and attacker engagement in the papers listed in Table 4.1

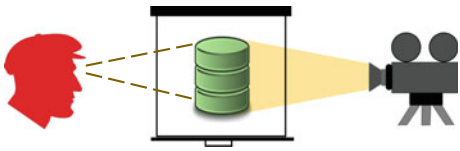


Fig. 4.6 We use the term *mimesis* to signify the creation of a specific false belief. In this example, a defender makes an adversary believe that a database exists when, in fact, it does not exist

Fig. 4.7 We use the term *crypsis* to refer to preventing an adversary from acquiring a true belief. In this example, a defender hides the true existence of a database from an adversary

guage and a set of basic concepts about which the security community can develop a shared understanding" [113]. The taxonomy that we define in the next two sections works towards this goal.

4.3 Taxonomy: Prescriptive Modeling Approach

A taxonomy is "a collection of controlled vocabulary terms organized into a hierarchical structure" in which each term "is in one or more parent/child (broader/narrower) relationships" [114]. Traditionally, each parent is called a *genus*, and each child is called a *species* [115]. Species are demarcated from their genus by a *specific difference*. The purpose of our taxonomy is to identify appropriate game-theoretic models for each species[2] of deception, so the specific differences that we use are related to game theory. These are: *private information* Θ, *players* $\mathcal{P}$, *actions* $\mathcal{A}$, and *time-horizon* $\mathcal{T}$.

[2]The most fine-grained species in a taxonomy are sometimes called the *infimae species*. Hence, we will sometimes refer to "types" of deception more formally as infimae species.

4.3.1 Private Information

One of these principles is private information. To deceive is to intentionally cause another agent either "to acquire or continue to have a false belief" or "to be prevented from acquiring or cease to have a true belief" [91]. These two categories are quite different. Figures 4.6 and 4.7 depict the two categories.

The first category involves instilling in another a specific falsity (Fig. 4.6). In the language of game theory, this requires the creation of a belief. In signaling games or partially observable stochastic games, agents maintain beliefs over the private information Θ of other agents. Deceptive actions manipulate these beliefs to create traps or decoys.

The second category of deception involves hiding a true belief (Fig. 4.7). This could be employed in cybersecurity to hide a vulnerability, or it could be deployed in privacy to protect sensitive information. In either case, the defender causes attackers to obtain noisy or uncertain information.

Biologists distinguish between inculcating false beliefs and hiding true information using the terms *crypsis* and *mimesis*. "In the former... an animal resembles some object which is of no interest to its enemy, and in doing so is concealed; in the latter... an animal resembles an object which is well known... and in so doing becomes conspicuous" [116]. We adopt these terms to signify the corresponding categories of cybersecurity deception.

4.3.2 Actors

The next specific difference is the set of actors or players $\mathcal{P}$ involved in deception. Based on the actors involved, cryptic deception can be divided into *intensive* and *extensive* deception. (See Figs. 4.8 and 4.9.)

Intensive deception modifies an actor (or its representation) in order to hide it from an adversary (Fig. 4.8). For example, some privacy techniques add noise to data about a user before publishing it. The user's own data is modified. By contrast, extensive deception uses a set of multiple users (or their representations) to hide information (Fig. 4.9). Examples in cyberspace include mix networks for anonymity.

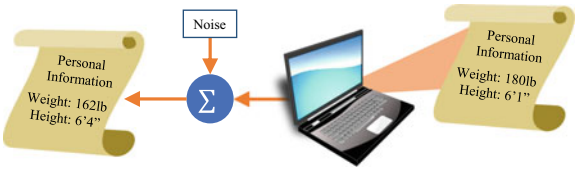


Fig. 4.8 Intensive deception. The defender alters the same object that is being hidden. In this example, the defender adds noise to private data

Fig. 4.9 Extensive deception. The defender hides an object using other objects in the environment. In this example, the defender dynamically changes the location of the private data

Many messages enter a network, where they are mixed in a chain of proxy servers before leaving the network. Any given message is hidden because of the presence of the other messages.

4.3.3 Actions

Deceivers can act in a number of ways, or actions $\mathcal{A}$. Within cryptic deception, we distinguish between deception that uses information and deception that uses motion. Deception that uses information tends to manipulate the data released about agents' properties, while deception that uses motion either modifies these properties over time or realizes these properties from a random variable. In other words, the first category is associated with creating noise, while the second category is associated with concepts such as agility and randomization.

4.3.4 Duration

Mimetic deception can be divided into static and dynamic scenarios. Dynamic games feature multiple interactions, while static games consist of only one interaction.[3] Currently, the majority of game-theoretic defensive deceptions use static models. The most popular set of mimetic deceptions that are static are honeypots, honeynets, and honeytokens. Dynamic models, however, are necessary in order to model sophisticated attacks such as advanced persistent threats (APTs). This is an important area for current cybersecurity research.

[3]Here, we consider one-shot games to be static even if they are not simultaneous-move games.

4.4 Taxonomy: Results

Figure 4.10 uses the specific differences of private information, actors, actions, and duration to create a taxonomy that breaks the genus of deception into multiple levels of species, terminating with the infimae species of perturbation, moving target defense, obfuscation, etc. Based on the taxonomy, Table 4.2 lists the definitions of each of the infimae species. The definitions are unambiguous as long as the specific differences are clearly defined. They are also mutually exclusive, because we have used mutually exclusive specific differences. Finally, the order of application of the specific differences does not matter. This implies that the taxonomy can be also represented using a binary lattice in four dimensions, one dimension for each of the specific differences. Interested readers can see [101] for a classification of the papers in Table 4.1 according to this taxonomy.

Section 4.2 made it clear that various game-theoretic models can be used to study each of the species of deception. Within these possibilities, however, it seems useful for us to highlight approaches that we think best capture the essential elements

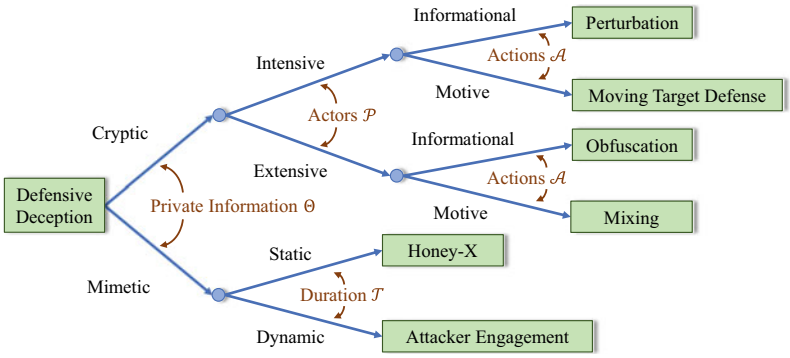


Fig. 4.10 Tree diagram breakdown of deception into various species. The specific differences correspond to the game-theoretic notions of private information, actors, actions, and duration. The infimae species are called perturbation, moving target defense, obfuscation, mixing, honey-x, and attacker engagement

Table 4.2 Infimae species and definition

Infimae species	Definition
Perturbation	Cryptic, intensive, informational deception
Moving target defense	Cryptic, intensive, motive deception
Obfuscation	Cryptic, extensive, informational deception
Mixing	Cryptic, extensive, motive deception
Honey-X	Mimetic, static deception
Attacker engagement	Mimetic, dynamic deception

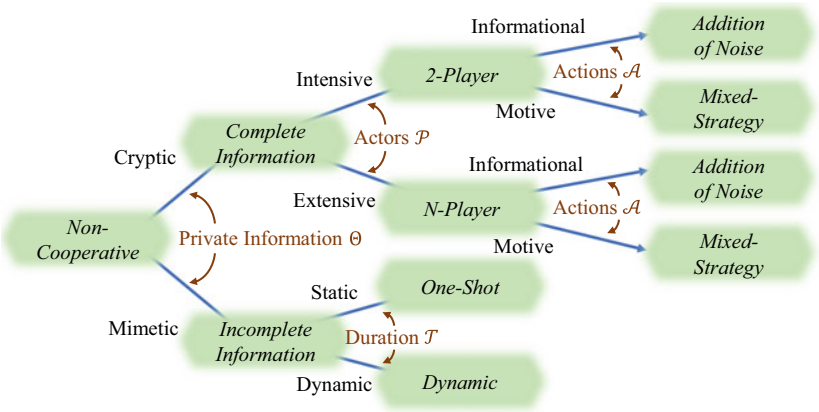


Fig. 4.11 Taxonomy with promising modeling approaches overlaid in a hierarchical fashion. For instance, cryptic, intensive, and motive deception (i.e., moving target defense) can be modeled by non-cooperative, complete-information, two-player games with mixed strategies. Each of the modeling approaches is shown in a blurred shape to indicate that alternate approaches are possible

of each species. Figure 4.11 depicts these modeling approaches. The root node is non-cooperative games.[4] Within non-cooperative games, games with incomplete information are well suited for mimetic deception, since they can explicitly model the false attacker beliefs that defenders attempt to inculcate. One-shot models such as signaling games model honey-x. Chapter 6 uses this approach. Dynamic models are necessary to effectively study attacker engagement. We use Markov decision processes for this purpose in Chap. 7.

Within cryptic deception, two players are often sufficient to study intensive deception, in which a single defender attempts to directly deceive an attacker. By contrast, extensive deception involves multiple defenders who compete against each other in addition to competing against an attacker (e.g., Chap. 5).

In perturbation and obfuscation, actions often consist of choosing the variance of noise to add to published data (e.g., Chap. 5). In mixing and moving target defense, actions often consist of choosing mixed strategies. Through these mixed strategies, defenders probabilistically choose routing paths or network configurations.

4.5 Looking Forward

The next three chapters of this book present models for three of the species of defensive deception described by Figs. 4.10 and 4.11. Chapter 5 studies obfuscation using a complete-information, N-player game in which actors hide their private informa-

[4]Future research can consider cooperative games, but most current studies use non-cooperative models.

tion through the addition of noise. Chapter 6 studies honey-x using incomplete-information, one-shot games. Finally, Chap. 7 models attacker engagement. As an initial step towards dynamic games, the chapter augments a Stackelberg game with a Markov decision process. We refer readers interested in perturbation, moving target defense, and mixing to the survey article [101], which reviews 11 research works from these three species of deception.

4.6 Notes

Surveys of game theory and cybersecurity in general can be found in [25, 117, 118]. Deception in the context of military applications has been well studied by Barton Whaley and coauthors [29, 30, 119]. Some of these works include taxonomies. Specifically, [119] makes a distinction between "hiding the real" and "showing the false" which we build upon in our taxonomy. Scott Gerwehr and Russell Glenn add that some deceivers aim to inject noise: randomness or intense activity that slows an adversary's ability to act. We find analogous aims in perturbation or obfuscation for the purposes of privacy. They also distinguish between static and dynamic deception. We adopt this distinction, since it motivates modeling efforts using one-shot or multiple-interaction games [120].

This chapter is also related to [121], which categorizes methods of deception according to techniques such as impersonation, delays, fakes, camouflage, false excuses, and social engineering. Finally, the present taxonomy can be compared and contrasted to those developed by Kristin Heckman et al. [122]. One of these taxonomies breaks down malicious deception by stages including design of a cover story, planning, execution, monitoring, etc. Another distinguishes between defensive deceptions based on whether they enable defensive visibility or restrict attacker visibility.

Finally, several other works have contributed to the effort to create a taxonomy of deception. Michael Handel makes a distinction between deceptions targeted against capabilities and those against intentions [120]. Oltramari and coauthors define an ontology of cybersecurity based on human factors [123]. Finally, Neil Rowe [124] describes a taxonomy of deception in cyberspace based on linguistic theory.

Chapter 5
Obfuscation

As data ecosystems grow in size due to the IoT, researchers are developing obfuscation techniques that issue fake search engine queries, undermine location tracking algorithms, or evade government surveillance. These techniques raise two conflicts: one between each user and the machine learning algorithms which track the users, and one between the users themselves. This chapter captures the first conflict with a Stackelberg game and the second conflict with a mean-field game. We combine both into a dynamic and strategic bi-level framework. Besides game theory, the nature of obfuscation demands that we draw upon techniques from machine learning and differential privacy.

5.1 Introduction to Obfuscation

As discussed in Chap. 4, obfuscation is the addition of noise by a group of actors to valuable data in order to hide trends within the data. Based on the terminology used in that chapter (c.f., Fig. 4.10), obfuscation is cryptic, extensive, and informational. This suggests that it is well-modeled using complete-information, N-player games that quantify utility based on a player's ability to extract a pattern or signal from noisy data (c.f., Fig. 4.11).

Finn and Nissenbaum describe two obfuscation technologies: *CacheCloak* and *TrackMeNot* [125]. *TrackMeNot* is a browser extension that generates randomized search queries in order to prevent trackers from assembling accurate profiles of their users [37]. In the realm of the IoT, *CacheCloak* provides a way for a user to access location-based services without revealing his or her exact geographical position [17]. The app predicts multiple possibilities for the path of a user, and then retrieves location-based information for each path. An adversary tracking the requests is left with many possible paths rather than a unique one. As another example, the browser extension *ScareMail* adds words relevant to terrorism to every email that a user issues, postulating that wide adoption of this technique would make dragnet

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_5

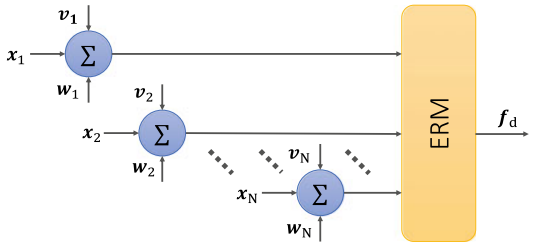


Fig. 5.1 Data flow in the obfuscation-tracking model. Users $1, \ldots, N$ have data x_i with labels y_i. They add noise $v_i \sim \mathcal{V}_i$ to x_i, and the learner promises noise $w_i \overset{\text{i.i.d.}}{\sim} \mathcal{W}$. Noise degrades accuracy but improves privacy. The users and the learner have misaligned incentives

surveillance difficult [126]. Designers of these obfuscation technologies see them as a way for users to resist tracking and put pressure on machine learning algorithms to guarantee some privacy protection.

This chapter uses game theory to identify conditions under which the threat of user obfuscation motivates machine learners to promise privacy protection. We construct a bi-level framework to model this interaction. At the user level, a large number of users play a Mean-Field Game (MFG) (c.f. [127]) to decide whether to use obfuscation. At the learner level, a machine learner plays a Stackelberg game [81] to decide whether to promise some level of privacy protection in order to avoid obfuscation by the users.

5.2 Model

Figure 5.1 depicts an interaction between a set of users $i \in \mathbb{S} = \{1, \ldots, N\}$ and a learner L. Users submit possibly perturbed data to L, and L releases a statistic or predictor f_d of the data. Assume that the data generating process is a random variable $\mathcal{Z}$ with a fixed but unknown distribution. Denote the realized data by $z_i \overset{\text{i.i.d.}}{\sim} \mathcal{Z}$, $i \in \mathbb{S}$. Each data point is composed of a feature vector $x_i \in \mathbb{R}^d$ and a label $y_i \in \{-1, 1\}$. The goal of the learner L is to predict y_i given x_i, based on the trained classifier or predictor f_d. Table 5.1 summarizes the notation for this chapter.

We investigate whether it is advantageous for L to promise some level of privacy protection in order to avoid user obfuscation.[1] L adds noise with the same variance to each data point x_i. For $i \in \mathbb{S}$, $k \in 1, \ldots, d$, L draws $w_i^{(k)} \overset{\text{i.i.d}}{\sim} \mathcal{W}$, where $\mathcal{W}$ is a mean-zero Gaussian random variable with standard deviation a_L. While differential privacy (c.f., Sect. 5.2.2) often considers Laplace noise, we use Gaussian noise for reasons of mathematical convenience. Knowing a_L, each user adds noise $v_i^{(k)} \overset{\text{i.i.d.}}{\sim} \mathcal{V}_i$,

[1] L can accomplish this by collecting data at low resolution. This is consistent with the spirit of differential privacy (c.f., Sect. 5.2.2), in which a learner publishes ϵ_p.

Table 5.1 Notation for this chapter

Notation	Meaning
$i \in \mathbb{F} = \{1, \ldots, N\}$, L	Users (followers) and learner
f_d, f^*	Learned and reference predictors
$z_i \overset{\text{i.i.d.}}{\sim} \mathcal{Z}$, $i \in \mathbb{F}$	Data
$x_i \in \mathbb{R}^d$ and $y_i \in \{-1, 1\}$, $i \in \mathbb{F}$	Feature vector and label
w_i, $w_i^{(k)} \overset{\text{i.i.d}}{\sim} \mathcal{W}$, $k \in 1, \ldots, d$, $i \in \mathbb{F}$	Noise added by learner
v_i, $v_i^{(k)} \overset{\text{i.i.d.}}{\sim} \mathcal{V}_i$, $k \in 1, \ldots, d$, $i \in \mathbb{F}$	Noise added by user
a_L, a_F^i, $i \in \mathbb{F}$	Standard deviation of learner and user noise
$\tilde{x}_i \triangleq x_i + v_i + w_i$, $i \in \mathbb{F}$.	Perturbed data points
$e_g\left(a_L, \bar{a}_F^{-i}, a_F^i\right)$	Accuracy level as a function of perturbation
$e_p\left(a_L, a_F^i\right)$	Privacy level as a function of perturbation
$u_L(a_L, \bar{a}_F)$	Learner utility function
$u_F^i(a_L, \bar{a}_F^{-i}, a_F^i)$	User utility function
A_L, A_F^i, P_F^i, C_L, C_F^i	Utility function weights
$P(a_L)$ and $AC(a_L, \bar{a}_F^{-i})$	Utility function meta-parameters
$BR_F\left(\bar{a}_F^{-i} \mid a_L\right)$	Best response of the users to the learner
$\Gamma\left(a_L\right)$	User perturbation induced by the learner
$(a_L^\dagger, a_F^{1\dagger}, a_F^{2\dagger}, \ldots, a_F^{N\dagger})$	Equilibrium strategies
τ, κ	Perturbation threshold, meta-parameter

$k \in 1, \ldots, d$, where $\mathcal{V}_i$ is Gaussian with variance $(a_S^i)^2$. The perturbed data points are given by $\tilde{x}_i = x_i + v_i + w_i$, $i \in \mathbb{S}$.

Machine learning scenarios online and in the IoT feature large numbers of users. *Mean-field games* [127] model such scenarios, in which agents respond to statistics of the actions of other agents. Individual actions do not have large impacts on the mean field. Therefore, define $\bar{a}_S^2 = \frac{1}{N}\sum_{i=1}^N (a_S^i)^2$, the average variance of the perturbations of every user, and $(\bar{a}_S^{-i})^2 = \frac{1}{N}\sum_{j=1}^N (a_S^j)^2 - \frac{1}{N}(a_S^i)^2$, the average variance of the perturbations of every user *other than* i. Learner L reacts to $\bar{a}_S$. Each individual user i reacts to both a_L and $\bar{a}_S^{-i}$, since the average perturbation of the other users affects the result that he receives from the learner.

5.2.1 Empirical Risk Minimization

Empirical Risk Minimization (ERM) refers to one popular family of machine learning. In ERM, L calculates a value of an output $f_d \in F$ that minimizes the empirical risk, i.e., the total penalty due to imperfect classification of the realized data. Define a *loss function* $l\left(\tilde{z}_i, f\right)$, which expresses the penalty due to a single perturbed data

point $\tilde{z}_i$ for the output f. L obtains f_d given by Eq. 5.1, where $\rho \geq 0$ is a constant and $R(f)$ is a regularization term to prevent overfitting:

$$f_d = \arg\min_{f \in F} \rho R(f) + \frac{1}{N} \sum_{i=1}^{N} l(\tilde{z}_i, f), \tag{5.1}$$

Expected loss provides a measure of the accuracy of the output of ERM. Let f^* denote the f which minimizes the expected loss for unperturbed data:

$$f^* = \operatorname*{argmin}_{f \in F} \mathbb{E}\{\rho R(f) + l(\mathcal{Z}, f)\}. \tag{5.2}$$

In Definition 5.1, f^* forms a reference to which the expected loss of the perturbed classifier f_d can be compared.

Definition 5.1 (ϵ_g-*Accuracy*) Let f_d and f^* denote the perturbed classifier and the classifier which minimizes expected loss, respectively. Let ϵ_g be a positive scalar. We say that f_d is ϵ_g-accurate if it satisfies

$$\mathbb{E}\{\rho R(f_d) + l(\mathcal{Z}, f_d)\} \leq \mathbb{E}\left\{\rho R\left(f^*\right) + l\left(\mathcal{Z}, f^*\right)\right\} + \epsilon_g. \tag{5.3}$$

Calculating the precise value of ϵ_g is beyond the scope of the current chapter. As an approximation, however, let us assume that the accuracy loss is linear in the total added perturbation, and inversely related to the number of nodes and regularization constant. This is summarized by Definition 5.2.

Definition 5.2 (*Accuracy Level*) If L perturbs with variance a_L^2, user $i \in \mathbb{S}$ perturbs with $(a_S^i)^2$, and the other users perturb with $(\bar{a}_S^{-i})^2$, then estimate the difference ϵ_g in expected loss between the perturbed classifier and the population-optimal classifier using the function e_g:

$$\epsilon_g \propto e_g\left(a_L, \bar{a}_S^{-i}, a_S^i\right) \triangleq \frac{1}{\rho^2 N}\left(a_L^2 + \frac{N-1}{N}\left(\bar{a}_S^{-i}\right)^2 + \frac{1}{N}\left(a_S^i\right)^2\right).$$

5.2.2 *Differential Privacy*

Using differential privacy, a machine learning agent promises a bound ϵ_p on the maximum information leaked about an individual. Let $\mathcal{A}(*)$ denote an algorithm and D denote a database. Let D' denote a database that differs from D by only one entry (e.g., the entry of the user under consideration). Let c be some set among all possible sets C in which the output of the algorithm $\mathcal{A}$ may fall. Then Definition 5.3 quantifies privacy using the framework of differential privacy [128, 129].

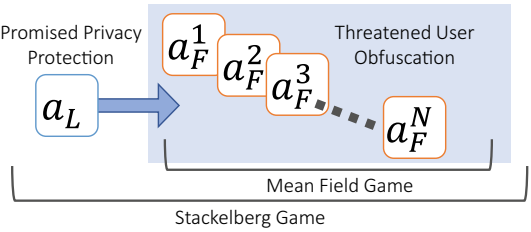


Fig. 5.2 Bi-level structure of the strategic interaction. Users may adopt obfuscation technologies in a cascading manner. This is modeled by an MFG. To avoid this, a learner can proactively add noise to their data. His interaction with the users in modeled by a Stackelberg game

Definition 5.3 (ϵ_p-*Privacy*) - An algorithm $\mathcal{A}(B)$ taking values in a set C provides (ϵ_p, δ)-differential privacy if, for all D, D' that differ in at most one entry, and for all $c \in C$,

$$\mathbb{P}\{\mathcal{A}(D) \in c\} \leq \exp\{\epsilon_p\}\, \mathbb{P}\left\{\mathcal{A}\left(D'\right) \in c\right\} + \delta. \tag{5.4}$$

For a cryptographically small δ, the degree of randomness determines the privacy level ϵ_p. Lower values of ϵ_p correspond to more privacy. That randomness is attained through the noise added in the forms of $\mathcal{V}$ and $\mathcal{W}$.

Lemma 5.4 (Privacy Level) *If L adds noise with variance a_L^2 and user $i \in \mathbb{S}$ perturbs with variance $(a_S^i)^2$, then the user obtains differential privacy level $\epsilon_p \in (0, 1)$ on the order of*

$$\epsilon_p \propto e_p\left(a_L, a_S^i\right) \triangleq \left(a_L^2 + \left(a_S^i\right)^2\right)^{-1/2}. \tag{5.5}$$

Proof In [130], Dwork and Roth present a differential privacy bound for the Gaussian Mechanism. Equation 5.5 solves this bound for ϵ_p [18].

5.2.3 *Bi-level Game*

Let $\mathbb{R}_M$ denote a subset $[0, M]$ of the nonnegative real numbers, and let M be arbitrarily large.[2] Let $a_L \in \mathbb{R}_M$ denote the standard deviation of the noise added by the learner. If the users are not satisfied with this level of privacy protection, they may add noise with standard deviations $a_S^i \in \mathbb{R}_M$, $i \in \mathbb{S}$ (Fig. 5.2).

Define a utility function by $u_L : \mathbb{R}_M^2 \to \mathbb{R}$ such that $u_L(a_L, \bar{a}_S)$ gives the utility that L receives for using noise a_L^2 while the users add an average noise of $\bar{a}_S^2$. Also define utility functions $u_S^i : \mathbb{R}_M^3 \to \mathbb{R}$ such that user $i \in \mathbb{S}$ receives utility $u_S^i(a_L, \bar{a}_S^{-i}, a_S^i)$ for obfuscating with variance $(a_S^i)^2$ while the other users obfuscate with average variance $(\bar{a}_S^{-i})^2$ and L perturbs with a_L^2. Reasonable u_L and u_S^i, $i \in \mathbb{S}$, are given by

[2] This rigorously deals with large perturbation variances.

$$u_L\left(a_L, \bar{a}_S\right) = A_L \exp\left\{-e_g\left(a_L, \bar{a}_S^{-i}, a_S^i\right)\right\} - C_L \mathbf{1}_{\{a_L>0\}},$$

$$u_S^i\left(a_L, \bar{a}_S^{-i}, a_S^i\right) = A_S^i \exp\left\{-e_g\left(a_L, \bar{a}_S^{-i}, a_S^i\right)\right\} - P_S^i\left(1 - \exp\left\{-e_p\left(a_L, a_S^i\right)\right\}\right) - C_S^i \mathbf{1}_{\{a_S^i>0\}},$$

where A_L (resp. A_S^i) gives the maximum benefit to the learner (resp. to each user) for output accuracy, P_S^i gives the maximum privacy loss to each user, and C_L (resp. C_S^i) gives the flat cost of perturbation for the learner (resp. to each user).

5.2.4 Equilibrium Requirements

Chronologically, L first promises perturbation a_L, and then the users choose obfuscation a_S^i, $i \in \mathbb{S}$. The solution, however, proceeds backwards in time.

Mean-Field Game

Given the promised a_L, the group of users plays an MFG in which each user best responds to the average perturbation of the other users.[3] Consider symmetric utility functions for the users, that is, $A_S^i = A_S$, $P_S^i = P_S$, and $C_S^i = C_S$, $i \in \mathbb{S}$. Let $BR_S : \mathbb{R}_M \to \mathbb{R}_M$ denote a best response function such that

$$BR_S\left(\bar{a}_S^{-i} \mid a_L\right) = \underset{a_S^i \in \mathbb{R}_M}{\arg\max}\, u_S^i\left(a_L, \bar{a}_S^{-i}, a_S^i\right) \tag{5.6}$$

gives the set of best responses for user $i \in \mathbb{S}$ to the average perturbation $\bar{a}_S^{-i}$ of the other users, given that the learner has promised a_L^2. Then the equilibrium of the MFG is $a_S^{1*} = a_S^{2*} = \ldots = a_S^{N*}$ (which is also equal to $\bar{a}_S^*$) which satisfies the fixed-point equation

$$\bar{a}_S^* \in BR_S\left(\bar{a}_S^* \mid a_L\right). \tag{5.7}$$

Now define a mapping $\Gamma : \mathbb{R}_M \to \mathbb{R}_M$ such that $\Gamma(a_L)$ gives the $\bar{a}_S^*$ which satisfies Eq. (5.7) given[4] a_L. We say that, by promising a_L, L *induces* $\bar{a}_S^* = \Gamma(a_L)$.

Stackelberg Game

Since L promises a_L^2 before the users obfuscate, L is a Stackelberg leader, and the users are collectively a Stackelberg follower which plays $\Gamma(a_L)$. The optimality equation for L is

$$a_L^* \in \underset{a_L \in \mathbb{R}_M}{\arg\max}\, u_L\left(a_L, \Gamma\left(a_L\right)\right). \tag{5.8}$$

[3] This is a strategic interaction, because each user would prefer to protect her own privacy while making use of accurate data from the other users.

[4] We will apply a selection criteria to ensure there is only one $\bar{a}_S^*$.

Definition 5.5 (*Perfect Bayesian Nash Equilibrium*) A perfect Bayesian Nash equilibrium (c.f., [90]) of the overall game is $(a_L^\dagger, a_S^{1\dagger}, a_S^{2\dagger}, \ldots, a_S^{N\dagger})$ such that $\bar{a}_S^\dagger = a_S^{1\dagger} = a_S^{2\dagger} = \cdots = a_S^{N\dagger}$, and

$$\bar{a}_S^\dagger = \Gamma\left(a_L^\dagger\right) = BR_S\left(\bar{a}_S^\dagger \,|\, a_L^\dagger\right), \tag{5.9}$$

$$a_L^\dagger \in \underset{a_L \in \mathbb{R}_M}{\arg\max}\, u_L\left(a_L, \Gamma\left(a_L\right)\right). \tag{5.10}$$

5.3 Mean-Field Game Analysis

First, Lemma 5.6 characterizes BR_S.

Lemma 5.6 (Best Response) *Define* $AC(a_L, \bar{a}_S^{-i}) \triangleq A_S \exp\{-e_g(a_L, \bar{a}_S^{-i}, 0)\} + C_S$ *and* $P(a_L) \triangleq P_S(1 - \exp\{-e_p(a_L, 0)\})$. *Then* BR_S *is given by*

$$BR_S\left(\bar{a}_S^{-i} \,|\, a_L\right) = \begin{cases} 0, & \text{if } P\left(a_L\right) < AC\left(a_L, \bar{a}_S^{-i}\right) \\ M, & \text{if } P\left(a_L\right) > AC\left(a_L, \bar{a}_S^{-i}\right) \\ [0, M], & \text{if } P\left(a_L\right) = AC\left(a_L, \bar{a}_S^{-i}\right) \end{cases}.$$

Figure 5.3 depicts Lemma 5.6. Users with low privacy sensitivity (left) never obfuscate, while users with high privacy sensitivity (right) always obfuscate. Importantly, users with moderate privacy sensitivity (center) cascade: each user i obfuscates if $\bar{a}_S^{-i}$ is high. Theorem 5.7 states that the MFG equilibria occur at the fixed points of the best response mappings.

Theorem 5.7 (MFG Equilibrium) *Given a promised privacy protection level* $a_L^\dagger$, *Eq. (5.9) is satisfied by the symmetric strategies* $a_S^{\dagger 1} = \ldots = a_S^{\dagger N} = \bar{a}_S^\dagger$, *where* $\bar{a}_S^\dagger = \Gamma\left(a_L^\dagger\right)$

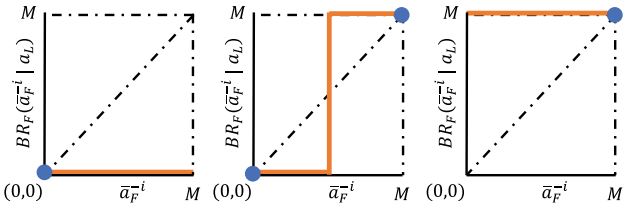


Fig. 5.3 Best response mappings (orange) for each user i against the other users $-i$. MFG equilibria occur at the intersections (blue circles) of the mappings with the identity mapping

$$= \begin{cases} 0, & \text{if } P(a_L) < AC(a_L, M) < AC(a_L, 0) \\ \{0, M\} & \text{if } AC(a_L, M) \leq P(a_L) \leq AC(a_L, 0) \\ M & \text{if } AC(a_L, M) < AC(a_L, 0) < P(a_L) \end{cases}.$$

In the middle case, u_S^i is higher for $\bar{a}_S^\dagger = 0$ than for $\bar{a}_S^\dagger = M$. Therefore, we select $\bar{a}_S^\dagger = 0$ and write $\Gamma(a_L) = M\mathbf{1}_{\{P(a_L) > AC(a_L, 0)\}}$.

5.4 Stackelberg Game

Next, L chooses a_L in order to maximize $u_L(a_L, \Gamma(a_L))$.

5.4.1 Status Quo Equilibrium

Lemma 5.8 characterizes a scenario in which L does not perturb.

Lemma 5.8 (Status Quo Stackelberg game Scenario) *If $P_S - C_S < A_S$, then $\Gamma_L(0) = 0$. In this case, the optimal $a_L^\dagger = 0$, for which L receives his maximum possible utility: $u_L(0, 0) = A_L$.*

$P_S - C_S < A_S$ holds if users are willing to suffer a total loss of privacy in order to obtain complete accuracy. We have called this the *status quo* because it seems to represent the current preferences of many users.

5.4.2 Equilibrium Outside of the Status Quo

Consider $P_S - C_S > A_S$. Define $\tau \in \mathbb{R}_M$ such that $P(\tau) = AC(\tau, 0)$. By promising to perturb with at least τ, L is able to induce $\Gamma(\tau) = 0$, i.e., to make it incentive-compatible for the users to not obfuscate. But we must analyze whether promising τ is incentive-compatible for L. Since the analytical expression for τ is cumbersome, define an approximation $\hat{\tau} > \tau$, where $\hat{\tau}^2 = 1/\ln\{P_S/(P_S - C_S)\}$. Next, define $\kappa \triangleq 1/(\rho^2 N)$. Then $u_L(a_L, \Gamma(a_L))$ is

$$\approx \begin{cases} 0, & \text{if } a_L^2 = 0 \\ -C_L, & \text{if } 0 < a_L^2 < \hat{\tau}^2 \\ A_L \exp\left\{-\kappa a_L^2\right\} - C_L, & \text{if } \hat{\tau}^2 \leq a_L^2 < M \end{cases}.$$

u_L is maximized by either 0 or $\hat{\tau}$ according to Theorem 5.9.

Table 5.2 Equilibrium results of the bi-level game

Parameter regime	$\bar{a}_S^\dagger$	$a_L^\dagger$
(1) $P_S - C_S < A_S$	0	0
(2) $P_S - C_S > A_S \bigcap \frac{1}{\rho^2 N} > \ln\left\{\frac{A_L}{C_L}\right\} \ln\left\{\frac{P_S}{P_S - C_S}\right\}$	M	0
(3) $P_S - C_S > A_S \bigcap \frac{1}{\rho^2 N} < \ln\left\{\frac{A_L}{C_L}\right\} \ln\left\{\frac{P_S}{P_S - C_S}\right\}$	0	$\hat{\tau}$

Theorem 5.9 (Stackelberg Game Equilibrium) *For $P_S - C_S > A_S$, the perturbation promise which satisfies Eq. (5.10) is*

$$a_L^\dagger = \begin{cases} 0, & \text{if } \frac{1}{\rho^2 N} > \ln\left\{\frac{A_L}{C_L}\right\} \ln\left\{\frac{P_S}{P_S - C_S}\right\} \\ \hat{\tau}, & \text{if } \frac{1}{\rho^2 N} < \ln\left\{\frac{A_L}{C_L}\right\} \ln\left\{\frac{P_S}{P_S - C_S}\right\} \end{cases}. \tag{5.11}$$

Theorem 5.9 shows that high costs C_S of user perturbation incentivize L to promise privacy protection, because users easily decide not to obfuscate. On the other hand, high privacy sensitivity P_S decreases L's incentive to add noise. Somewhat surprisingly, high accuracy sensitivity A_L leads L to promise privacy protection.[5]

5.4.3 Summary of Results

Table 5.2 summarizes the results of the overall game. The equilibrium strategies $\bar{a}_S^\dagger$ and $a_L^\dagger$ satisfy Definition 5.5. Equilibrium 1 is the *status quo* equilibrium in which users submit unperturbed data and L does not protect it. This equilibrium achieves complete accuracy at the cost of complete loss of privacy. In Equilibrium 2, users obfuscate as much as possible. L lacks the incentive to promise privacy protection, so he does not perturb. He receives zero utility, making machine learning useless. Equilibrium 3 is the best equilibrium. In this scenario, *the threat* of user obfuscation convinces L to promise privacy protection $a_L^2 = \hat{\tau}^2$. The users accept this level, and do not adopt obfuscation.

5.5 Discussion of Results

Privacy skeptics argue that users are not willing to pay for privacy protection. This is captured by $P_S - C_S < A_S$, which leads to Equilibrium 1. But as obfuscation technologies such as *TrackMeNot* [37] and *CacheCloak* [17] continue to develop, the cost C_S of obfuscation will decrease, and the awareness P_S of privacy concerns will

[5] Accuracy sensitivity increases his sensitivity to user obfuscation.

increase. Both will lead to $P_S - C_S > A_S$. In Equilibrium 3, obfuscation motivates the learner to promise some level of privacy protection. Nevertheless, some scenarios cannot be improved by obfuscation. In the case of Equilibrium 2, users perturb their data as much as possible, but this only decreases the opportunities for meaningful analysis and discourages machine learning.

5.6 Related Work

Related work includes research in *privacy markets*, in which a learner pays users to report data truthfully [131, 132]. In [133, 134], users play a multiple person, prior-commitment game, which determines how much they obfuscate. In these papers, the learner calculates the average of a dataset. Finally, [18] considers a Stackelberg game, but it does not include a mean-field interaction among the users.

Chapter 6
Honey-X

The previous chapter discussed obfuscation, in which the defender's goal is to hide valuable information within noise. Obfuscation, in other words, is a species of crypsis (Sect. 4.3). But in other species of deception, the defender aims to create a specific false belief. This is called mimesis. The present chapter studies static mimesis, or, *honey-x*, which takes its name from technologies related to honeypots, honeytokens, etc.

Signaling games (Sect. 3.2) provide a promising game-theoretic approach to study honey-x. But signaling games inherently model honey-x that is undetectable. In this chapter, we extend signaling games by including a detector that gives off probabilistic warnings when the sender acts deceptively. Then we derive pooling and partially separating equilibria of the game. We find that (1) high-quality detectors eliminate some pure-strategy equilibria, (2) detectors with high true-positive rates encourage more honest signaling than detectors with low false-positive rates, (3) receivers obtain optimal outcomes for equal-error-rate detectors, and (4) surprisingly, deceptive senders sometimes benefit from highly accurate deception detectors. We illustrate these results with an application to defensive deception for network security.

6.1 Introduction to Honey-X

Recent years have witnessed a surge in the deployment of honeypots to detect and counteract adversarial behaviors. Honeypots have been used in large commercial enterprises and government agencies. They provide a high quality of alerts with a low demand for human and computational resources.

The idea of honeypots has been extended to honeyfiles and honeynets. Honeyfiles serve as bait for an attacker to access so that his movements can be detected. They are simple to implement and have been shown to provide reliable detection. In [103],

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_6

honeyfiles are used as the key mechanism to manage compliance and protect an enterprise from insider threats. Honeynets are networks of honeypots used to gather additional attack information, including the motives and strategies of the attackers, the communication patterns among the attacks, and the timing of the attacks. The Honeypot Project [105] is a global security research organization started in 1999 that develops open-source honeynet tools. Honeypots, honeyfiles, and honeynets are instances of the type of deception that we call honey-x.[1]

6.1.1 Signaling Games for Mimesis

As described in Chap. 4, honey-x is deception that is mimetic and static. Hence, it can be captured by models that are (1) information asymmetric and (2) one-shot.[2] In this species of deception, one party (hereafter, the *sender*) possesses private information unknown to the other party (hereafter, the *receiver*). Based on her private information, the sender communicates a possibly untruthful message to the receiver. Then the receiver forms a belief about the private information of the sender, and chooses an action. The players act *strategically*, in the sense that they each seek a result that corresponds to their individual incentives.

Cheap-talk signaling games [38] model interactions that are strategic, dynamic, and information-asymmetric. In cheap-talk signaling games, a sender S with private information communicates a message to a receiver R, who acts upon it. Then both players receive utility based on the private information of S and the action of R. Recently, signaling games have been used to model deceptive interactions in resource allocation [109], network defense [50, 135], and cyber-physical systems [136].

6.1.2 Cost and Detection in Signaling Games

The phrase *cheap talk* signifies that the utility of both players is independent of the message that S communicates to R. In cheap-talk signaling games, therefore, there is no cost or risk of lying per se. Truth-telling sometimes emerges in equilibrium, but not through the penalization or detection of lying. We can say that cheap-talk signaling games model deception that is *undetectable* and *costless.*

In economics literature, Navin Kartik has proposed a signaling game model that rejects the second of these two assumptions [39]. In Kartik's model, S pays an explicit cost to send a message that does not truthfully represent her private information.

[1] Since honeynets allow dynamic interaction with the attacker, some honeynets could quality as attacker engagement (Chap. 7).

[2] By one shot, we mean that the interaction between the players is not repeated, although the interaction is dynamic in the sense that one player transmits a signal and the other player acts after he observes the signal. If the interaction is repeated, we call the deception *attacker engagement*, studied in Chap. 7.

This cost could represent the effort required, e.g., to obfuscate data, to suppress a revealing signal, or to fabricate misleading data. In equilibrium, the degree of deception depends on the lying cost. Contrary to cheap-talk games, Kartik's model studies deception that has a cost. Yet the deception is still undetectable.

In many scenarios, however, deception can be detected with some probability. Consider the issue of so-called *fake news* in social media. Fake news stories about the 2016 U.S. Presidential Election reportedly received more engagement on Facebook than news from real media outlets [137]. In the wake of the election, Facebook announced its own program to detect fake news and alert users about suspicious articles. As another example, consider deceptive product reviews in online marketplaces. Linguistic analysis has been used to detect this *deceptive opinion spam* [138]. Finally, consider the deployment of honeypots as a technology for defensive deception. Attackers have developed tools that detect the virtual machines often used to host honeypots.

Therefore, we propose a model of signaling games in which a detector emits probabilistic evidence of deception. The detector can be interpreted in two ways. It can be understood as a technology that R uses to detect deception, such as a phishing detector in an email client. Alternatively, it can be understood as the inherent tendency of S to emit cues to deception when she misrepresents her private information. For instance, in interpersonal deception, lying is cognitively demanding, and sometimes liars give off cues from these cognitive processes [34]. R uses the evidence from the detector to form a belief about whether S's message honestly conveys her private information.

Our work focuses mostly on detectable deception for the cheap-talk scenario, based on the observation that the cost of signaling is often significantly less than the value of a successful deception. Examples of such applications include military scenarios such as the feint of the Allied armies at Pas de Calais before the D-Day attack at Normandy, economic applications such as the set of three unenforced announcements of a "last chance" early retirement program at the University of California, and security scenarios such as signs which claim that surveillance cameras are observing customers in a market[3] [139]. Readers interested in the same model with small lying costs can see [140].

6.1.3 Model Overview

Let us consider an informal overview of our model. S has a private bit of information that is either 0 or 1. S can either transmit the true bit to R, or S can run a program that deceptively sends the opposite bit to R. With some probability, however, R can

[3] Deception in all of these cases requires some expense, but the cost is little compared to the utility gain from successful deception. For example, the difference for the Allies between winning and losing at Normandy was likely much higher than the cost of making misleading preparations to invade at Pas de Calais.

detect the running of the deceptive program. In that case, we say that the detector gives off an alarm. The detector is imperfect and can also give off false alarms. Based on the bit sent by S and the evidence emitted by the detector (alarm or no alarm), R attempts to guess the private bit. R receives a higher utility if he is correct than if he is wrong, and S receives a higher payoff if R is wrong than if he is correct.

Naturally, several questions arise. What is the equilibrium of the game? How, if at all, is the equilibrium different than that of a traditional cheap-talk signaling game? Is detection always harmful to the sender? Finally, how does the design of the detector affect the equilibrium? This chapter addresses each of these questions.

Section 6.2 describes our model and equilibrium concept. In Sect. 6.3, we find pure-strategy and mixed-strategy equilibria. Then Sect. 6.4 evaluates the sensitivity of the equilibria to changes in detector characteristics. We prove that detectors that prioritize high true-positive rates encourage more honest equilibrium behavior than detectors that prioritize low false-positive rates. Section 6.5 describes an application to defensive deception for network security. Finally, we discuss the implications of our results in Sect. 6.6.

6.2 Model

Signaling games are two-player games between a sender (S) and a receiver (R). These games are *information asymmetric*, because S possesses information that is unknown to R. They are also *dynamic*, because the players' actions are not simultaneous. S transmits a message to R, and then R acts upon the message. Generally, signaling games can be *non-zero-sum*, which means that the objectives of S and R are not direct negations of each other's objectives. Figure 6.1 depicts the traditional signaling game between S and R, augmented by a detector block. We call this augmented signaling game a *signaling game with evidence*.

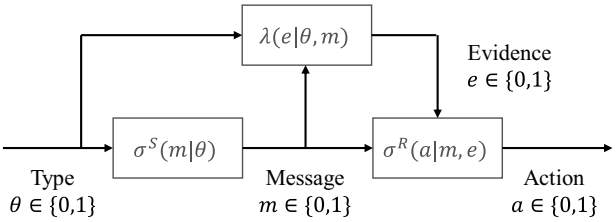


Fig. 6.1 Signaling games with evidence add the red detector block to the S and R blocks. The probability $\lambda(e \mid \theta, m)$ of emitting evidence e depends on S's type θ and the message m that she transmits

Table 6.1 Summary of notation

Notation	Meaning
$S,\ R$	Sender and receiver
$\theta \in \Theta,\ m \in \mathbb{M},\ a \in \mathbb{A}$	Types, messages, actions
$e \in \mathbb{EV}$	Detector evidence
$u^X(\theta, m, a)$	Utility functions of player $X \in \{S, R\}$
$p(\theta)$	Prior probability of θ
$\sigma^S(m \mid \theta) \in \Gamma^S$	Mixed strategy of S of type θ
$\lambda(e \mid \theta, m)$	Probability of e for type θ & message m
$\sigma^R(a \mid m, e) \in \Gamma^R$	Mixed strategy of R given $m,\ e$
$\mu^R(\theta \mid m, e)$	Belief of R that S is of type θ
$U^S\left(\sigma^S, \sigma^R \mid \theta\right)$	Expected utility for S of type θ
$U^R\left(\sigma^R \mid \theta, m, e\right)$	Expected utility for R given $\theta,\ m,\ e$
$\alpha \in [0, 1]$	"Size" of detector (false-positive rate)
$\beta \in [\alpha, 1]$	"Power" of detector (true positive rate)
$\Delta_0^R,\ \Delta_1^R$	Utility function meta-parameters
$J,\ G$	Detector meta-parameters
$\tau(J, G, p)$	Truth induction rate
$\tilde{U}^S \in \mathbb{R}$ and $\tilde{U}^R \in \mathbb{R}$	A priori expected equilibrium utilities of S and R

6.2.1 Types, Messages, Evidence, Actions, and Beliefs

We consider binary information and action spaces in order to simplify analysis. Table 6.1 summarizes the notation. Let $\theta \in \Theta = \{0, 1\}$ denote the private information of S. Signaling games refer to θ as a *type*. The type could represent, e.g., whether the sender is a malicious or benign actor, whether she has one set of preferences over another, or whether a given event observable to S but not to R has occurred. The type is drawn[4] from a probability distribution p, where $\sum_{\theta \in \Theta} p(\theta) = 1$ and $\forall \theta \in \Theta,\ p(\theta) \geq 0$.

Based on θ, S chooses a message $m \in \mathbb{M} = \{0, 1\}$. She may use mixed strategies, i.e., she may select each m with some probability. Let $\sigma^S \in \Gamma^S$ denote the strategy of S, such that $\sigma^S(m \mid \theta)$ gives the probability with which S sends message m given that she is of type θ. The space of strategies satisfies

$$\Gamma^S = \left\{ \sigma^S \mid \forall \theta,\ \sum_{m \in \mathbb{M}} \sigma^S(m \mid \theta) = 1;\ \forall \theta, m,\ \sigma^S(m \mid \theta) \geq 0 \right\}.$$

Since Θ and $\mathbb{M}$ are identical, a natural interpretation of an honest message is $m = \theta$, while a deceptive message is represented by $m \neq \theta$.

[4] Harsanyi conceptualized type selection as a randomized move by a non-strategic player called *nature* (in order to map an incomplete information game to one of complete information) [84].

Next, the detector emits evidence based on whether the message m is equal to the type θ. The detector emits $e \in \mathbb{EV} = \{0, 1\}$ by the probability $\lambda(e \mid \theta, m)$. Let $e = 1$ denote an *alarm* and $e = 0$ *no alarm*. Note that e is emitted with an exogenous probability that R does not control. In this respect, the detector can be seen as a second move by nature. The probability with which a detector records a true positive is called the *power* $\beta \in [0, 1]$ of the detector. For simplicity, we set both true-positive rates to be equal: $\beta = \lambda(1 \mid 0, 1) = \lambda(1 \mid 1, 0)$. Similarly, let α denote the *size* of the detector, which refers to the false-positive rate. We have $\alpha = \lambda(1 \mid 0, 0) = \lambda(1 \mid 1, 1)$. A valid detector has $\beta \geq \alpha$. This is without loss of generality, because otherwise α and β can be relabeled.

After receiving both m and e, R chooses an action $a \in \mathbb{A} = \{0, 1\}$. R may also use mixed strategies. Let $\sigma^R \in \Gamma^R$ denote the strategy of R such that his mixed-strategy probability of playing action a given message m and evidence e is $\sigma^R(a \mid m, e)$. The space of strategies is $\Gamma^R =$

$$\left\{\sigma^R \mid \forall m, e, \sum_{a \in \mathbb{A}} \sigma^R(a \mid m, e) = 1;\ \forall e, m, a,\ \sigma^R(a \mid m, e) \geq 0\right\}.$$

Based on m and e, R forms a belief about the type θ of S. In mimetic deception (c.f., Sect. 4.3.1), S tries to induce in R a specific false belief. For all θ, m, and e, define $\mu^R : \Theta \to [0, 1]$ such that $\mu^R(\theta \mid m, e)$ gives the likelihood with which R believes that S is of type θ given message m and evidence e. R uses belief μ^R to decide which action to chose.

6.2.2 Utility Functions

Let $u^S : \Theta \times \mathbb{M} \times \mathbb{A} \to \mathbb{R}$ denote a utility function for S such that $u^S(\theta, m, a)$ gives the utility that she receives when her type is θ, she sends message m, and R plays action a. Similarly, let $u^R : \Theta \times \mathbb{M} \times \mathbb{A}$ denote R's utility function so that $u^R(\theta, m, a)$ gives his payoff under the same scenario.

Table 6.2 gives the assumptions necessary to characterize a deceptive interaction. Assumption 1 is that u^S and u^R do not depend (exogenously) on m, i.e., the interaction is a cheap-talk game. Assumptions 2–3 state that R receives higher utility if he correctly chooses $a = \theta$ than if he chooses $a \neq \theta$. Finally, Assumptions 4–5 state that S receives higher utility if R chooses $a \neq \theta$ than if he chooses $a = \theta$. Together, Assumptions 1–5 characterize a *cheap-talk signaling game with evidence*.

Define an expected utility function $U^S : \Gamma^S \times \Gamma^R \to \mathbb{R}$ such that $U^S(\sigma^S, \sigma^R \mid \theta)$ gives the expected utility to S when she plays strategy σ^S, given that she is of type θ. This expected utility is given by

Table 6.2 Signaling game utility function assumptions

Number	Assumption
1	$\forall\theta \in \Theta,\ \forall a \in \mathbb{A}$ $u^R(\theta, 0, a) = u^R(\theta, 1, a)$ $u^S(\theta, 0, a) = u^S(\theta, 1, a)$
2	$\forall m, \tilde{m} \in \mathbb{M},\ u^R(0, m, 0) > u^R(0, \tilde{m}, 1)$
3	$\forall m, \tilde{m} \in \mathbb{M},\ u^R(1, m, 0) < u^R(1, \tilde{m}, 1)$
4	$\forall m, \tilde{m} \in \mathbb{M},\ u^S(0, m, 0) < u^S(0, \tilde{m}, 1)$
5	$\forall m, \tilde{m} \in \mathbb{M},\ u^S(1, m, 0) > u^S(1, \tilde{m}, 1)$

$$U^S\left(\sigma^S, \sigma^R \mid \theta\right) = \sum_{a\in\mathbb{A}} \sum_{e\in\mathbb{EV}} \sum_{m\in\mathbb{M}} \sigma^R(a \mid m, e)\, \lambda(e \mid \theta, m)\, \sigma^S(m \mid \theta)\, u^S(\theta, m, a)\,. \tag{6.1}$$

Note that in Eq. (6.1), R's mixed-strategy probability $\sigma^R(a \mid m, e)$ of playing action a depends on the evidence e that he observes. S must anticipate the probability of leaking evidence e. She does this using $\lambda(e \mid \theta, m)$. This probability is *exogenous*, not controlled by either player. The probability does not appear in the standard signaling-game without evidence.

Next define $U^R : \Gamma^R \to \mathbb{R}$ such that $U^R(\sigma^R \mid \theta, m, e)$ gives the expected utility to R when he plays strategy σ^R given message m, evidence e, and sender type θ. The expected utility function is given by

$$U^R(\sigma^R \mid \theta, m, e) = \sum_{a\in\mathbb{A}} \sigma^R(a \mid m, e)\, u^R(\theta, m, a)\,.$$

6.2.3 Equilibrium Concept

In two-player games, Nash equilibrium defines a strategy profile in which each player best responds to the optimal strategies of the other player [80]. Signaling games motivate the extension of Nash equilibrium in two ways. First, information asymmetry requires R to maximize his expected utility over the possible types of S. An equilibrium in which S and R best respond to each other's strategies given some belief μ^R is called a *Bayesian Nash equilibrium* [84]. We also require R to update μ^R rationally. *Perfect Bayesian Nash equilibrium* (PBNE) captures this constraint. Definition 6.1 applies PBNE to our game.

Definition 6.1 (*Perfect Bayesian Nash Equilibrium* [90]) A PBNE of a cheap-talk signaling game with evidence is a strategy profile $(\sigma^{S*}, \sigma^{R*})$ and posterior beliefs $\mu^R(\theta \mid m, e)$ such that

$$\forall \theta \in \Theta,\ \sigma^{S*} \in \arg\max_{\sigma^S \in \Gamma^S} U^S\left(\sigma^S, \sigma^{R*} \mid \theta\right), \tag{6.2}$$

$\forall m \in \mathbb{M},\ \forall e \in \mathbb{EV}$,

$$\sigma^{R*} \in \arg\max_{\sigma^R \in \Gamma^R} \sum_{\theta \in \Theta} \mu^R\left(\theta \mid m, e\right) U^R(\sigma^R \mid \theta, m, e), \tag{6.3}$$

and if $\sum_{\tilde{\theta} \in \Theta} \lambda(e \mid \tilde{\theta}, m)\sigma^S(m \mid \tilde{\theta}) p(\tilde{\theta}) > 0$, then

$$\mu^R\left(\theta \mid m, e\right) = \frac{\lambda\left(e \mid \theta, m\right) \mu^R\left(\theta \mid m\right)}{\sum_{\tilde{\theta} \in \Theta} \lambda\left(e \mid \tilde{\theta}, m\right) \mu^R\left(\tilde{\theta} \mid m\right)}, \tag{6.4}$$

where

$$\mu^R\left(\theta \mid m\right) = \frac{\sigma^S\left(m \mid \theta\right) p\left(\theta\right)}{\sum_{\hat{\theta} \in \Theta} \sigma^S\left(m \mid \hat{\theta}\right) p\left(\hat{\theta}\right)}. \tag{6.5}$$

If $\sum_{\tilde{\theta} \in \Theta} \lambda(e \mid \tilde{\theta}, m)\sigma^S(m \mid \tilde{\theta}) p(\tilde{\theta}) = 0$, then $\mu^R\left(\theta \mid m, e\right)$ may be set to any probability distribution over Θ.

Equations (6.4)–(6.5) require the belief to be set according to Bayes' Law. First, R updates her belief according to m using Eq. (6.5). Then R updates her belief according to e using Eq. (6.4). This step is not present in the traditional definition of PBNE for signaling games.

There are three categories of PBNE: *separating*, *pooling*, and *partially separating* equilibria. These are defined based on the strategy of S. In separating PBNE, the two types of S transmit opposite messages. This allows R to infer S's type with certainty. In pooling PBNE, both types of S send messages with identical probabilities. That is, $\forall m \in \mathbb{M},\ \sigma^S(m \mid 0) = \sigma^S(m \mid 1)$. This makes m useless to R. R updates his belief based only on the evidence e. Equations (6.4)–(6.5) yield

$$\mu^R\left(\theta \mid m, e\right) = \frac{\lambda\left(e \mid \theta, m\right) p\left(\theta\right)}{\sum_{\tilde{\theta} \in \Theta} \lambda\left(e \mid \tilde{\theta}, m\right) p\left(\tilde{\theta}\right)}. \tag{6.6}$$

In partially separating PBNE, the two types of S transmit messages with different, but not completely opposite, probabilities. In other words, $\forall m \in \mathbb{M},\ \sigma^S(m \mid 0) \neq \sigma^S(m \mid 1)$, and $\sigma^S(m \mid 0) \neq 1 - \sigma^S(m \mid 1)$. Equations (6.4)–(6.5) allow R to update his belief, but the belief remains uncertain.

6.3 Equilibrium Results

In this section, we find the PBNE of the cheap-talk signaling game with evidence. We present the analysis in four steps. In Sect. 6.3.1, we solve the optimality condition for R, which determines the structure of the results. In Sect. 6.3.2, we solve the optimality condition for S, which determines the equilibrium beliefs μ^R. We present the pooling equilibria of the game in Sect. 6.3.3. Some parameter regimes do not admit any pooling equilibria. For those regimes, we derive partially separating equilibria in Sect. 6.3.4.

First, Lemma 6.2 notes that one class of equilibria is not supported.

Lemma 6.2 *Under Assumptions 1-5, the game admits no separating PBNE.*

Section 6.8.1 gives the proof. Lemma 6.2 results from the opposing utility functions of S and R. S wants to deceive R, and R wants to correctly guess the type. It is not incentive-compatible for S to fully reveal the type by choosing a separating strategy.

6.3.1 Prior Probability Regimes

Next, we look for pooling PBNE. Consider R's optimal strategies σ^{R*} in this equilibrium class. Note that if the sender uses a pooling strategy on message m (i.e., if S with both $\theta = 0$ and $\theta = 1$ send message m), then $\sigma^{R*}(1 \mid m, e)$ gives R's optimal action a after observing evidence e. Messages do not reveal anything about the type θ, and R updates his belief using Eq. (6.6). For brevity, define the following notations:

$$\Delta_0^R \triangleq u^R\left(\theta = 0, m, a = 0\right) - u^R\left(\theta = 0, m, a = 1\right), \tag{6.7}$$

$$\Delta_1^R \triangleq u^R\left(\theta = 1, m, a = 1\right) - u^R\left(\theta = 1, m, a = 0\right). \tag{6.8}$$

Δ_0^R gives the benefit to R for correctly guessing the type when $\theta = 0$, and Δ_1^R gives the benefit to R for correctly guessing the type when $\theta = 1$. Since the game is a cheap-talk game, these benefits are independent of m. Lemmas 6.3–6.6 solve for σ^{R*} within five regimes of the prior probability $p(\theta)$ of each type $\theta \in \{0, 1\}$. Recall that $p(\theta)$ represents R's belief that S has type θ *before* R observes m or e.

Lemma 6.3 *For pooling PBNE, R's optimal actions σ^{R*} for evidence e and messages m on the equilibrium path*[5] *vary within five regimes of $p(\theta)$. The top half of Fig. 6.2 lists the boundaries of these regimes for detectors in which $\beta < 1 - \alpha$, and the bottom half of Fig. 6.2 lists the boundaries of these regimes for detectors in which $\beta > 1 - \alpha$.*

[5] In pooling PBNE, the message "on the equilibrium path" is the one that is sent by both types of S. Messages "off the equilibrium path" are never sent in equilibrium, although determining the actions that R *would play* if S were to transmit a message off the path is necessary in order to determine the existence of equilibria.

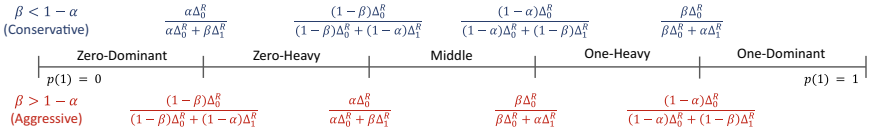


Fig. 6.2 PBNE differ within five prior probability regimes. In the Zero-Dominant regime, $p(\theta = 1) \approx 0$, i.e., type 0 dominates. In the Zero-Heavy regime, $p(\theta = 1)$ is slightly higher, but still low. In the middle regime, the types are mixed almost evenly. The One-Heavy regime has a higher $p(\theta = 1)$, and the One-Dominant regime has $p(\theta = 1) \approx 1$. The definitions of the regime boundaries depend on whether the detector is conservative or aggressive

Proof See Sect. 6.8.2.

Remark 6.4 Boundaries of the equilibrium regimes differ depending on the relationship between β and $1-\alpha$. β is the true-positive rate and $1-\alpha$ is the true-negative rate. Let us call detectors with $\beta < 1-\alpha$ *conservative* detectors, detectors with $\beta > 1-\alpha$ *aggressive detectors*, and detectors with $\beta = 1-\alpha$ *equal-error-rate* (*EER*) detectors. Aggressive detectors have high true-positive rates but also high false-positive rates. Conservative detectors have low false-positive rates but also low true-positive rates. Equal-error-rate detectors have an equal rate of false-positives and false-negatives.

Remark 6.5 The regimes in Fig. 6.2 shift toward the right as Δ_0^R increases. Intuitively, a higher $p(1)$ is necessary to balance out the benefit to R for correctly identifying a type $\theta = 0$ as Δ_0^R increases. The regimes shift toward the left as Δ_1^R increases for the opposite reason.

Lemma 6.6 gives the optimal strategies of R in response to pooling behavior within each of the five parameter regimes.

Lemma 6.6 *For each regime, σ^{R*} on the equilibrium path is listed in Table 6.3 if $\beta < 1-\alpha$ and in Table 6.4 if $\beta > 1-\alpha$. The row labels correspond to the Zero-Dominant (O-D), Zero-Heavy (0-H), Middle, One-Heavy (1-H), and One-Dominant (1-D) regimes.*

Proof See Sect. 6.8.2.

Remark 6.7 In the Zero-Dominant and One-Dominant regimes of all detector classes, R determines σ^{R*} based only on the overwhelming prior probability of one type over the other.[6] In the Zero-Dominant regime, R chooses $\sigma^{R*}(1 \mid m, e) = 0$

[6]For instance, consider an application to product reviews in an online marketplace. A product may be low ($\theta = 0$) or high ($\theta = 1$) quality. A reviewer (S) may describe the product as poor ($m = 0$) or as good ($m = 1$). Based on the wording of the review, a reader (R) may be suspicious ($e = 1$) that the review is fake, or he may not be suspicious ($e = 0$). He can then buy ($a = 1$) or do not buy ($a = 0$) the product. According to Remark 6.7, if R has a strong prior belief that the product is high quality ($p(1) \approx 1$), then he will ignore both the review m and the evidence e, and he will always buy the product ($a = 1$).

Table 6.3 $\sigma^{R*}(1\,|m, e)$ in pooling PBNE with $\beta < 1 - \alpha$

	$\sigma^{R*}(1\,\|\,0, 0)$	$\sigma^{R*}(1\,\|\,0, 1)$	$\sigma^{R*}(1\,\|\,1, 0)$	$\sigma^{R*}(1\,\|\,1, 1)$
0-D	0	0	0	0
0-H	0	1	0	0
Middle	0	1	1	0
1-H	1	1	1	0
1-D	1	1	1	1

Table 6.4 $\sigma^{R*}(1\,|m, e)$ in pooling PBNE with $\beta > 1 - \alpha$

	$\sigma^{R*}(1\,\|\,0, 0)$	$\sigma^{R*}(1\,\|\,0, 1)$	$\sigma^{R*}(1\,\|\,1, 0)$	$\sigma^{R*}(1\,\|\,1, 1)$
0-D	0	0	0	0
0-H	0	0	1	0
Middle	0	1	1	0
1-H	0	1	1	1
1-D	1	1	1	1

for all m and e, and in the One-Dominant regime, R chooses $\sigma^{R*}(1\,|\,m, e) = 1$ for all m and e.

Remark 6.8 In the Middle regime of both detector classes, R chooses[7] $\sigma^{R*}(1\,|\,m, 0) = m$ and $\sigma^{R*}(1\,|\,m, 1) = 1 - m$. In other words, R believes the message of S if $e = 0$ and does not believe the message of S if $e = 1$.

6.3.2 *Optimality Condition for S*

Next, we must check to see whether each possible pooling strategy is optimal for S. This depends on what R *would do* if S were to deviate and send a message off the equilibrium path. R's action in that case depends on his beliefs for messages off the path. In PBNE, these beliefs can be set arbitrarily. The challenge is to see whether beliefs μ^R exist such that each pooling strategy is optimal for both types of S. Lemmas 6.9–6.10 give conditions under which such beliefs exist.

Lemma 6.9 *Let m be the message on the equilibrium path. If $\sigma^{R*}(1\,|\,m, 0) = \sigma^{R*}(1\,|\,m, 1)$, then there exists a μ^R such that pooling on message m is optimal for both types of S. For brevity, let $a^* \triangleq \sigma^{R*}(1\,|\,m, 0) = \sigma^{R*}(1\,|\,m, 1)$. Then μ^R is given by*

[7] For the same application to online marketplaces as in footnote 6, if R does not have a strong prior belief about the quality of the product (e.g., $p(1) \approx 0.5$), then he will trust the review (play $a = m$) if $e = 0$, and he will not trust the review (he will play $a = 1 - m$) if $e = 1$.

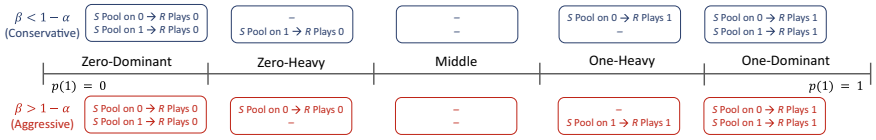


Fig. 6.3 PBNE in each of the parameter regimes defined in Fig. 6.2. For $m \in \{0, 1\}$, "S pool on m" signifies $\sigma^{S*}(m \mid 0) = \sigma^{S*}(m \mid 1) = 1$. For $a \in \{0, 1\}$, "R Plays a" signifies $\sigma^{R*}(a \mid 0, 0) = \sigma^{R*}(a \mid 0, 1) = \sigma^{R*}(a \mid 1, 0) = \sigma^{R*}(a \mid 1, 1) = 1$. Lemma 6.9 gives μ^R. The dominant regimes support pooling PBNE on both messages. The heavy regimes support pooling PBNE on only one message. The middle regime does not support any pooling PBNE

$$\forall e \in \mathbb{EV},\ \mu^R\left(\theta = a^* \mid 1 - m, e\right) \geq \frac{\Delta^R_{1-a^*}}{\Delta^R_{1-a^*} + \Delta^R_{a^*}}.$$

Lemma 6.10 *If $\sigma^{R*}(1 \mid m, 0) = 1 - \sigma^{R*}(1 \mid m, 1)$ and $\beta \neq 1 - \alpha$, then there does not exist a μ^R such that pooling on message m is optimal for both types of S.*

Proof See Sect. 6.8.3 for the proofs of Lemmas 6.9–6.10.

The implications of these lemmas can be seen in the pooling PBNE results that are presented next.

6.3.3 Pooling PBNE

Theorem 6.11 gives the pooling PBNE of the game.

Theorem 6.11 (Pooling PBNE) *The pooling PBNE are summarized by Fig. 6.3.*

Proof The theorem results from combining Lemmas 6.3–6.10, which give the equilibrium σ^{S*}, σ^{R*}, and μ^R.

Remark 6.12 For $\beta < 1 - \alpha$, the Zero-Heavy regime admits only a pooling PBNE on $m = 1$, and the One-Heavy regime admits only a pooling PBNE on $m = 0$. We call this situation (in which, typically, $m \neq \theta$) a *falsification convention*. (See Sect. 6.4.2.) For $\beta > 1 - \alpha$, the Zero-Heavy regime admits only a pooling PBNE on $m = 0$, and the One-Heavy regime admits only a pooling PBNE on $m = 1$. We call this situation (in which, typically, $m = \theta$) a *truth-telling convention*.

Remark 6.13 For $\beta \neq 1 - \alpha$, the Middle regime does not admit any pooling PBNE. This result is not found in conventional signaling games for deception, in which all regimes support pooling PBNE [135]. It occurs because R's responses to message m depends on e, i.e., $\sigma^{R*}(1 \mid 0, 0) = 1 - \sigma^{R*}(1 \mid 0, 1)$ and $\sigma^{R*}(1 \mid 1, 0) = 1 - \sigma^{R*}(1 \mid 1, 1)$. One of the types of S prefers to deviate to the message off the equilibrium path. Intuitively, for a conservative detector, S with type $\theta = m$ prefers

to deviate to message $1-m$, because his deception is unlikely to be detected. On the other hand, for an aggressive detector, S with type $\theta = 1-m$ prefers to deviate to message $1-m$, because his honesty is likely to produce a false-positive alarm, which will lead R to guess $a = m$. Section 6.8.3 includes a formal derivation of this result.

6.3.4 *Partially Separating PBNE*

For $\beta \neq 1-\alpha$, since the Middle regime does not support pooling PBNE, we search for partially separating PBNE. In these PBNE, S and R play mixed strategies. In mixed-strategy equilibria in general, each player chooses a mixed strategy that makes the other players indifferent between the actions that they play with positive probability. Theorems 6.14–6.15 give the results.

Theorem 6.14 (Partially-Separating PBNE for Conservative Detectors) *For $\beta < 1-\alpha$, within the Middle Regime, there exists an equilibrium in which the sender strategies are*

$$\sigma^{S*}(m=1\,|\,\theta=0) = \frac{\beta^2}{\beta^2-\alpha^2} - \frac{\alpha\beta\Delta_1^R}{\left(\beta^2-\alpha^2\right)\Delta_0^R}\left(\frac{p(1)}{1-p(1)}\right),$$
$$\sigma^{S*}(m=1\,|\,\theta=1) = \frac{\alpha\beta\Delta_0^R}{\left(\beta^2-\alpha^2\right)\Delta_1^R}\left(\frac{1-p(1)}{p(1)}\right) - \frac{\alpha^2}{\beta^2-\alpha^2},$$

the receiver strategies are

$$\sigma^{R*}(a=1\,|\,m=0,e=0) = \frac{1-\alpha-\beta}{2-\alpha-\beta},$$
$$\sigma^{R*}(a=1\,|\,m=0,e=1) = 1,$$
$$\sigma^{R*}(a=1\,|\,m=1,e=0) = \frac{1}{2-a-b},$$
$$\sigma^{R*}(a=1\,|\,m=1,e=1) = 0,$$

and the beliefs are computed by Bayes' Law in all cases.

Theorem 6.15 (Partially-Separating PBNE for Aggressive Detectors) *For any $g \in [0,1]$, let $\bar{g} \triangleq 1-g$. For $\beta > 1-\alpha$, within the Middle Regime, there exists an equilibrium in which the sender strategies are*

$$\sigma^{S*}(m=1\,|\,\theta=0) = \frac{\bar{\alpha}\bar{\beta}\Delta_1^R}{\left(\bar{\alpha}^2-\bar{\beta}^2\right)\Delta_0^R}\left(\frac{p(1)}{1-p(1)}\right) - \frac{\bar{\beta}^2}{\bar{\alpha}^2-\bar{\beta}^2},$$
$$\sigma^{S*}(m=1\,|\,\theta=1) = \frac{\bar{\alpha}^2}{\bar{\alpha}^2-\bar{\beta}^2} - \frac{\bar{\alpha}\bar{\beta}\Delta_0^R}{\left(\bar{\alpha}^2-\bar{\beta}^2\right)\Delta_1^R}\left(\frac{1-p(1)}{p(1)}\right),$$

the receiver strategies are

$$\sigma^{R*}(a=1\,|\,m=0,e=0) = 0,$$
$$\sigma^{R*}(a=1\,|\,m=0,e=1) = \frac{1}{\alpha+\beta},$$
$$\sigma^{R*}(a=1\,|\,m=1,e=0) = 1,$$
$$\sigma^{R*}(a=1\,|\,m=1,e=1) = \frac{\alpha+\beta-1}{\alpha+\beta},$$

and the beliefs are computed by Bayes' Law in all cases.

Proof See Sect. 6.8.4 for the proofs of Theorems 6.14–6.15.

Remark 6.16 In Theorem 6.14, S chooses the σ^{S*} that makes R indifferent between $a=0$ and $a=1$ when he observes the pairs $(m=0,e=0)$ and $(m=1,e=0)$. This allows R to choose mixed strategies for $\sigma^{R*}(1\,|\,0,0)$ and $\sigma^{R*}(1\,|\,1,0)$. Similarly, R chooses $\sigma^{R*}(1\,|\,0,0)$ and $\sigma^{R*}(1\,|\,1,0)$ that make both types of S indifferent between sending $m=0$ and $m=1$. This allows S to choose mixed strategies. A similar pattern follows in Theorem 6.15 for σ^{S*}, $\sigma^{R*}(1\,|\,0,1)$, and $\sigma^{R*}(1\,|\,1,1)$.

Remark 6.17 Note that none of the strategies are functions of the sender utility u^S. As shown in Sect. 6.4, this gives the sender's expected utility a surprising relationship with the properties of the detector.

Figures 6.4 and 6.5 depict the equilibrium strategies for S and R, respectively, for an aggressive detector. Note that the horizontal axis is the same as the horizontal axis in Figs. 6.2 and 6.3.

The Zero-Dominant and One-Dominant regimes feature two pooling equilibria. In Fig. 6.4, the sender strategies for the first equilibrium are depicted by the red and blue curves, and the sender strategies for the second equilibrium are depicted by the green and black curves. These are pure strategies, because they occur with probabilities of zero or one. The Zero-Heavy and One-Heavy regimes support only one pooling equilibria in each case.

The Middle regime of $p(1)$ features the partially separating PBNE given in Theorem 6.15. In this regime, Fig. 6.5 shows that R plays a pure strategy when $e=0$ and a mixed strategy when[8] $e=1$. The next section investigates the relationships between these equilibrium results and the parameters of the game.

[8]On the other hand, for conservative detectors R plays a pure strategy when $e=1$ and a mixed strategy when $e=0$.

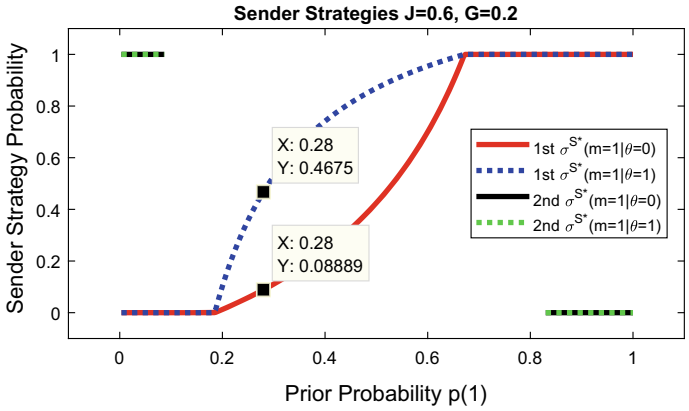


Fig. 6.4 Equilibrium sender strategies for $\beta = 0.9$, $\alpha = 0.3$, $\Delta_0^R = 15$, and $\Delta_1^R = 22$. The dominant regimes of $p(1)$ support both pooling on $m = 0$ and $m = 1$. The heavy regimes ($0.09 < p < 0.19$ and $0.67 < p < 0.83$) support only pooling on $m = 0$ and $m = 1$, respectively. The middle regime does not support any pooling PBNE, but does support a partially separating PBNE

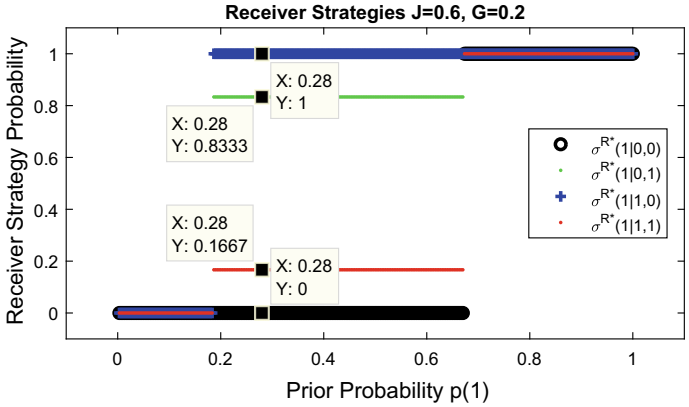


Fig. 6.5 Equilibrium receiver strategies for $\beta = 0.9$, $\alpha = 0.3$, $\Delta_0^R = 15$, and $\Delta_1^R = 22$. R plays pure strategies in both the dominant and heavy regimes. In the middle regime, two strategy components are pure, and two are mixed

6.4 Comparative Statics

In this section, we define quantities that we call the *quality* and *aggressiveness* of the detector. Then we define a quantity called *truth-induction*, and we examine the variation of truth-induction with the quality and aggressiveness of the detector.

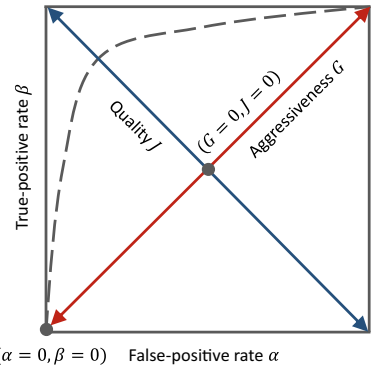


Fig. 6.6 The detector characteristics can be plotted in ROC-space. We also parameterize the detector characteristics by the orthogonal qualities $J \in [-1, 1]$ and $G \in [-1, 1]$. The dashed line gives a sample ROC-curve

6.4.1 Equilibrium Strategies Versus Detector Characteristics

Consider an alternative parameterization of the detector by the pair J and G, where $J = \beta - \alpha \in [-1, 1]$, and $G = \beta - (1 - \alpha) \in [-1, 1]$. J is called *Youden's J Statistic* [141]. Since an ideal detector has high β and low α, J parameterizes the *quality* of the detector. G parameterizes the *aggressiveness* of the detector, since an aggressive detector has $\beta > 1 - \alpha$ and a conservative detector has $\beta < 1 - \alpha$. Figure 6.6 depicts the transformation of the axes. Note that the pair (J, G) fully specifies the pair (α, β).

Figure 6.7 depicts the influences of J and G on S's equilibrium strategy. The red (solid) curves give $\sigma^{S*}(m = 1 \mid \theta = 0)$, and the blue (dashed) curves represent $\sigma^{S*}(m = 1 \mid \theta = 1)$. Although the Zero-Dominant and One-Dominant regimes support two pooling equilibria, Fig. 6.7 only plots one pooling equilibrium for the sake of clarity.[9]

In Column 1, J is fixed and G decreases from top to bottom. The top two plots have $G > 0$, and the bottom plot has $G < 0$. There is a regime change at exactly $G = 0$. At that point, the equilibrium $\sigma^{S*}(1 \mid 0)$ and $\sigma^{S*}(1 \mid 1)$ flip to their complements. Here a small perturbation in the characteristics of the detector leads to a large change in the equilibrium strategies. In Column 2, G is fixed, and J decreases from top to bottom. The detector is conservative: $G = -0.1$. Note that a large J leads to a large Middle regime, i.e., a large range of $p(1)$ for which S plays mixed strategies in equilibrium.

6.4.2 Truth-Induction

Consider the Middle regimes of the games plotted in Column 2. Note that the probabilities with which both types of S send $m = 1$ decrease as $p(1)$ increases. On the

[9] We chose the pooling equilibrium in which $\sigma^{S*}(1 \mid 0)$ and $\sigma^{S*}(1 \mid 1)$ are continuous with the partially separating $\sigma^{S*}(1 \mid 0)$ and $\sigma^{S*}(1 \mid 1)$ that are supported in the Middle regime.

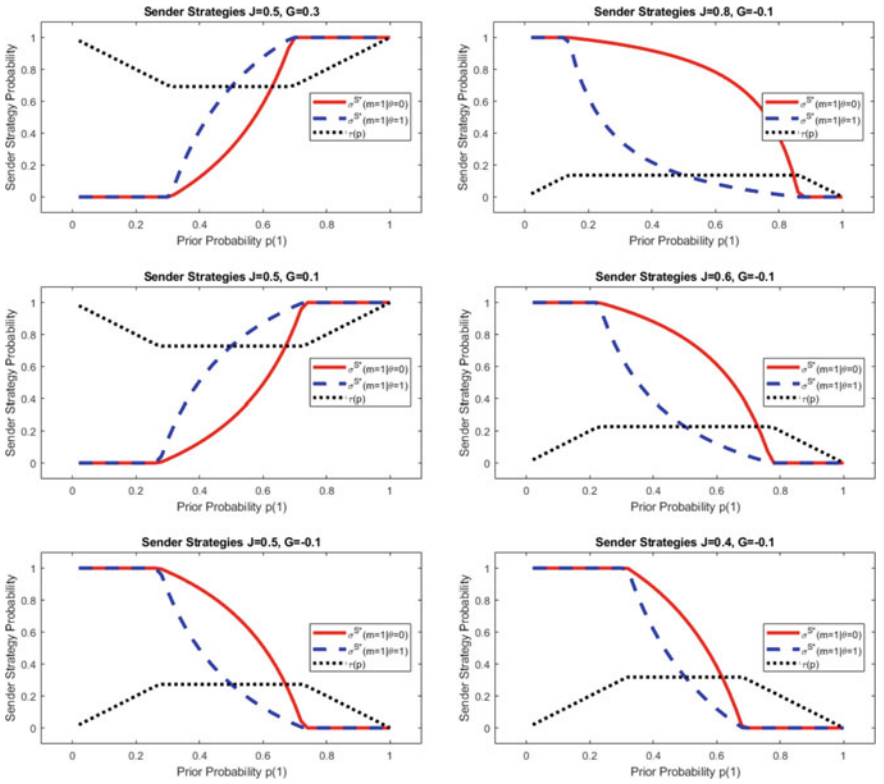


Fig. 6.7 Sender equilibrium strategies σ^{S*} and truth-induction rates τ. Column 1: detector quality J is fixed and aggressiveness G decreases from top to bottom. Column 2: G is fixed and J decreases from top to bottom. The sender equilibrium strategies are mixed within the middle regime, and constant within the other regimes

other hand, for games with aggressive detectors (c.f., Fig. 6.4), the probabilities with which both types of S send $m = 1$ increases as $p(1)$ increases. This suggests that S is somehow more "honest" for aggressive detectors.

In order to formalize this result, let $\sigma^{S*}(m \mid \theta; p)$ parameterize the sender's equilibrium strategy by the prior probability $p \triangleq p(1)$. Then define a mapping $\tau : [-1, 1]^2 \times [0, 1] \to [0, 1]$, such that $\tau(J, G, p)$ gives the *truth-induction rate* of the detector parameterized by (J, G) at the prior probability[10] p. We have

$$\tau(J, G, p) = \sum_{\theta \in \{0,1\}} p\sigma^{S*}(m = \theta \mid \theta; p). \tag{6.9}$$

[10]Feasible detectors have $J \leq 1 - |G|$. In addition, we only analyze detectors in which $\beta > \alpha$, which gives $J > 0$.

The quantity τ gives the proportion of messages sent by S for which $m = \theta$, i.e., for which S tells the truth. From this definition, we have Theorem 6.18.

Theorem 6.18 (Detectors and Truth Induction Rates) *Set $\Delta_0^R = \Delta_1^R$. Then, within prior probability regimes that feature unique PBNE (i.e., the Zero-Heavy, Middle, and One-Heavy Regimes), for all $J \in [0, 1]$ and $\forall p \in [0, 1]$, we have that*

$$\tau(J, G, p) \leq \frac{1}{2} \text{ for } G \in (-1, 0],$$
$$\tau(J, G, p) \geq \frac{1}{2} \text{ for } G \in [0, 1).$$

Proof See Sect. 6.8.5.

Remark 6.19 We can summarize Theorem 6.18 by stating that *aggressive detectors induce a truth-telling convention*, while *conservative detectors induce a falsification convention*.

The black curves in Fig. 6.7 plot $\tau(p)$ for each of the equilibrium strategies of S.

6.4.3 Equilibrium Utility

The a priori expected equilibrium utilities of S and R are the utilities that the players expect *before* θ is drawn. Denote these utilities by $\tilde{U}^S \in \mathbb{R}$ and $\tilde{U}^R \in \mathbb{R}$, respectively. For $X \in \{S, R\}$, the utilities are given by

$$\tilde{U}^X = \sum_{\theta \in \Theta} \sum_{m \in \mathbb{M}} \sum_{e \in \mathbb{EV}} \sum_{a \in \mathbb{A}} p(\theta) \, \sigma^{S*}(m \mid \theta) \, \lambda(e \mid \theta, m) \, \sigma^{R*}(a \mid m, e) \, u^X(\theta, m, a).$$

Figure 6.8 numerically illustrates the relationship between $\tilde{U}^R$ and detector quality J. R's utility improves as detector quality improves. In the Middle regime of this example, R's expected utility increases from $J = 0.2$ to $J = 0.8$. Intuitively, as his detection ability improves, his belief μ^R becomes more certain. In the Zero-Dominant, Zero-Heavy, One-Heavy, and One-Dominant regimes, R's equilibrium utility is not affected, because he ignores e and chooses a based only on prior probability.

Figure 6.9 numerically plots R's expected a priori equilibrium utility for various values of aggressiveness G. In the example, the same color is used for each detector with aggressiveness G and its opposite $-G$. Detectors with $G \geq 0$ are plotted using circles, and detectors with $G < 0$ are plotted using dashed lines. The utilities are the same for detectors with G and $-G$. In the Middle regime, expected utility increases as $|G|$ decreases from 0.5 to 0. This suggests that, for a fixed detector quality, it is optimal for R to use an EER detector.

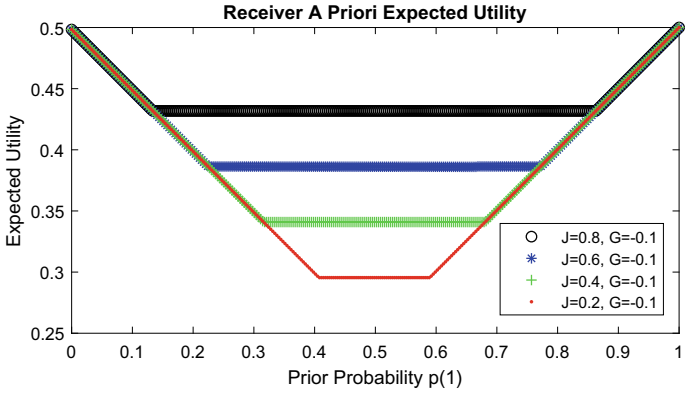


Fig. 6.8 R's a priori expected utility for varying J. Toward the extremes of $p(1)$, R's a priori expected utility does not depend on J, because R ignores e. But in the middle regime, R's expected utility increases with J

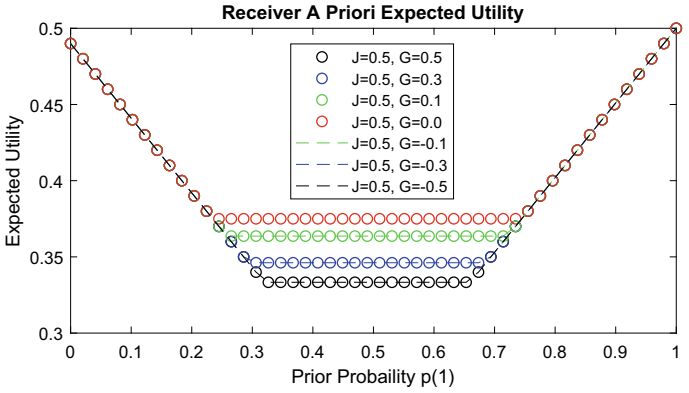


Fig. 6.9 R's a priori expected utility for varying G. Toward the extremes of $p(1)$, R's a priori expected utility does not depend on G, because R ignores e. But in the middle regime, R's expected utility decreases with $|G|$

6.5 Case Study

In this section, we apply signaling games with evidence to the use of defensive deception for network security. We illustrate the physical meanings of the action sets and equilibrium outcomes of the game for this application. We also present several results based on numerical experiments.

6.5.1 Motivation

Traditional cybersecurity technologies such as firewalls, cryptography, and Role-Based Access Control (RBAC) are insufficient against new generations of sophisticated and well-resourced adversaries. Attackers capable of APTs use techniques such as social engineering and hardware exploits to circumvent defenses and gain insider access. Stealthy and evasive maneuvers are crucial elements of APTs. Often, attackers are able to remain within a network for several months or years before they are detected by network defenders [142]. This gives the attackers an advantage of information asymmetry.

To counteract this information asymmetry, defenders have developed various technologies that detect attackers and provide attackers with false information. In fact, *honeypots* are classical mechanisms that achieve both goals. Honeypots are systems placed within a network in such a manner that the systems are never accessed by legitimate users. Any activity on the honeypots is evidence that the network has been breached by an attacker.

6.5.2 Signaling and Detection

Honeypots can roughly be divided into low-interaction and high-interaction, depending on the degree to which they simulate real systems. Low-interaction honeypots provide a simple interface to attackers. They do not simulate the functionality of production systems [97]. This low level of activity is a signal that the system is a honeypot. On the other hand, high-interaction honeypots enable a variety of user activities, such as interaction with real operating systems and SSH servers. This high level of activity is a deceptive signal: a signal that the system is a production system when it is really a honeypot.

This signal, however, is vulnerable to leaking evidence of deception. Such evidence may include irregular hardware components, uncharacteristic behavior, or hints of monitoring software [98, 99]. Thus, disguising a honeypot as a normal system by simulating high levels of activity is a type of detectable deception.[11]

6.5.3 Model Description

6.5.3.1 Players and Types

Figure 6.10 casts the honeypot interaction as a signaling game with evidence. The players are the *network defender* (the sender, S) and the *attacker* (the receiver, R).

[11] Interestingly, some research has also suggested artificially making normal systems appear to be honeypots, as a method of deterring attackers [100]. This is an opposite form of deceptive signaling, and can also be detected.

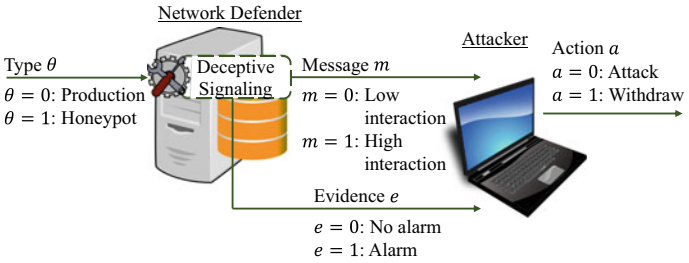


Fig. 6.10 A defender adds a production system or a honeypot to a network. Without any deception, the interaction level of the system is $m = \theta$. The defender can artificially alter the interaction level of the system, for example, by using a high-interaction honeypot. But this deception may be detected

The private information of the sender is the type of the system, $\theta \in \{0, 1\}$, where $\theta = 0$ represents a *production system*, and $\theta = 1$ represents a *honeypot*.

6.5.3.2 Messages

In general, the space of possible messages from a system to the attacker is large: it can include having or lacking an SSH server, using a real or virtual machine, receiving high or low amounts of incoming traffic, or occupying a central or peripheral place in a network. Both honeypots and production systems could have any of these settings. The settings constitute a message, because they influence the attacker's belief about whether a node is a production system or a honeypot. Here, we focus on one particular dimension of these settings: whether the system is low-interaction or high-interaction. Let $m = 1$ denote that a system allows only low levels of interaction, and let $m = 0$ denote that a system allows high levels of interaction.

Consider, for instance, installing a honeypot ($\theta = 1$) into a network. The simplest type of honeypot is low-interaction ($m = 1$). Low-interaction honeypots send an honest signal, since they have $m = \theta$. This tends to reveal that the systems are honeypots. On the other hand, honeypots that are high-interaction ($m = 0$) send a deceptive signal, since $m = 1 - \theta$. They appear to be production systems. Production systems ($\theta = 0$), of course, normally allow high-interaction ($m = 0$), since they are designed for normal user activities. But a network defender could attempt to decrease the available services on a production system ($m = 1$), in order to make an attacker suspect that the system is a honeypot ($\theta = 1$). This is also a deceptive signal.

6.5.3.3 Evidence

Both types of deceptive signals ($\theta = 1$, $m = 0$ and $\theta = 0$, $m = 1$) can be detected with some probability. For instance, if S makes a honeypot ($\theta = 1$) appear highly interactive ($m = 0$), R may observe traces of suspicious evidence such as "irregular

hardware components, uncharacteristic behavior (e.g., large system call latencies) and hints of monitoring software" [97–99]. These constitute an *alarm*: $e = 1$. If R does not notice these suspicious signals, then there is *no alarm*: $e = 0$. Similarly, if S creates programs or user policies for a production system ($\theta = 0$) that limit the available interactions ($m = 1$), then R may observe evidence that user behaviors are being artificially manipulated. This constitutes an alarm: $e = 1$. If R does not notice this manipulation, then there is no alarm: $e = 0$.

6.5.3.4 Actions

After observing the activity level m and leaked evidence e, R chooses an action $a \in \{0, 1\}$. Let $a = 0$ denote *attack*, and let $a = 1$ denote *withdraw*. Notice that R prefers to choose $a = \theta$, while S prefers that R choose $a \neq \theta$. Hence, Assumptions 2–5 are satisfied. If the cost of running the deceptive program is negligible, then Assumption 1 is also satisfied. This is reasonable if the programs are created once and deployed multiple times.

6.5.4 Equilibrium Results

We set the utility functions according to [50, 135]. Consider a detector with a true-positive rate $\beta = 0.9$ and a false-positive rate $\alpha = 0.3$. This detector has $\beta > 1 - \alpha$, so it is an aggressive detector, and the boundaries of the equilibrium regimes are given in the bottom half of Fig. 6.2. For this application, the Zero-Dominant and Zero-Heavy regimes can be called the Production-Dominant and Production-Heavy regimes, since type $\theta = 0$ represents a production system. Similarly, the One-Heavy and One-Dominant regimes can be called the Honeypot-Heavy and Honeypot-Dominant regimes.

We have plotted the equilibrium strategies for these parameters in Figs. 6.4 and 6.5 in Sect. 6.3. In the Production-Dominant regime ($p(\theta = 1) < 0.09$), there are very few honeypots. In equilibrium, S can set both types of systems to a high activity level ($m = 0$) or set both types of systems to a low activity level ($m = 1$). In both cases, regardless of the evidence e, R attacks. Next, for the Production-Heavy regime ($0.09 < p(1) < 0.19$), the only equilibrium strategy for S is a pooling strategy in which she leaves production systems ($\theta = 0$) active ($m = 0$) and runs a program on honeypots ($\theta = 1$) in order to make them appear active ($m = 1$) as well. R is able to detect ($e = 1$) the deceptive program on honeypots with probability $\beta = 0.9$, yet the prior probability $p(1)$ is low enough that it is optimal for R to ignore the evidence and attack ($a = 1$).

The Middle regime covers prior probabilities $0.19 < p(1) < 0.67$. The figures display the players' mixed strategies at $p(1) = 0.28$. For honeypots, S's optimal strategy is to leave the activity level low with approximately 50% probability ($\sigma^{S*}(1 \mid 1) \approx 0.47$) and to simulate a high activity level with approximately 50%

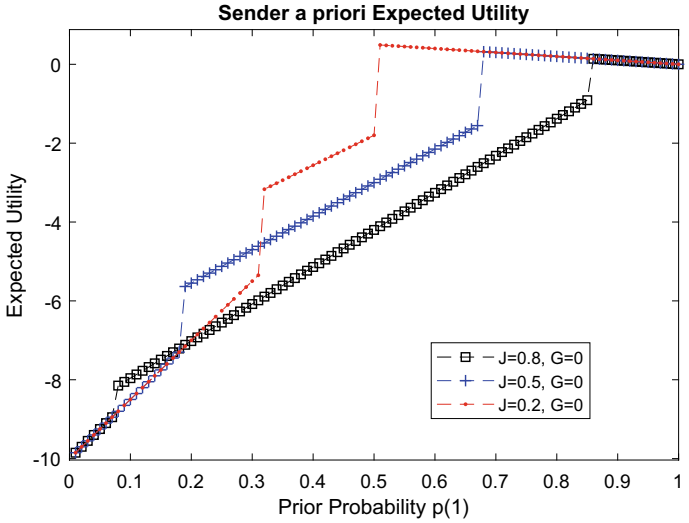


Fig. 6.11 S's a priori expected utility for varying J. From least accurate detector to the most accurate detector, the curves are colored red, blue, and black. Surprisingly, for some p, S does better with more accurate detectors than with less accurate detectors

probability. For production systems, S's optimal strategy is to decrease the activity level with a low probability ($\sigma^{S*}(1 \mid 0) \approx 0.09$) and to leave the activity level high with the remaining probability.

In the Middle regime, the receiver plays according to the activity level—i.e., trusts S's message—if $e = 0$. If $e = 1$ when $m = 0$, then most of the time, R does not trust S ($\sigma^{R*}(1 \mid 0, 1) \approx 0.83$). Similarly, if $e = 1$ when $m = 1$, then most of the time, R does not trust S ($\sigma^{R*}(1 \mid 1, 1) \approx 0.17$). The pattern of equilibria in the Honeypot-Heavy and Honeypot-Dominant regimes is similar to the pattern of equilibria in the Production-Heavy and Production-Dominant regimes.

6.5.5 Numerical Experiments and Insights

6.5.5.1 Equilibrium Utility of the Sender

Next, Corollary 6.20 considers the relationship between S's a priori equilibrium utility and the quality J of the detector.

Corollary 6.20 *Fix an aggressiveness G and prior probability $p(1)$. S's a priori expected utility $\tilde{U}^S$ is not necessarily a decreasing function of the detector quality J.*

Proof Figure 6.11 provides a counter-example.

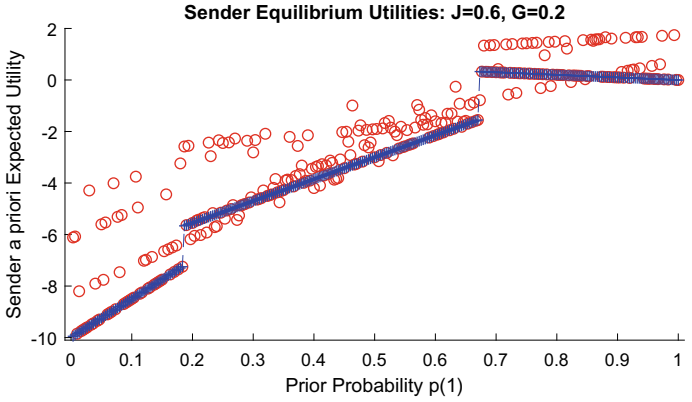


Fig. 6.12 S's a priori expected utility for playing her equilibrium strategy against (1) R's equilibrium strategy (in blue crosses), and (2) suboptimal strategies of R (in red circles). Deviations from R's equilibrium strategy almost always increase the expected utility of S

Corollary 6.20 is counterintuitive, because it seems that the player who attempts deception (S) should prefer poor deception detectors. Surprisingly, this is not always the case. Figure 6.11 displays the equilibrium utility for S in the honeypot example for three different J. At $p(1) = 0.4$, S receives the highest expected utility for the lowest quality deception detector. But at $p(1) = 0.1$, S receives the highest expected utility for the highest quality deception detector. In general, this is possible because the equilibrium regimes and strategies (e.g., in Theorems 6.11–6.15) depend on the utility parameters of R, not S.

In particular, this occurs because of an asymmetry between the types. S does very poorly if for a production system ($\theta = 0$), she fails to deceive R, so R attacks ($a = 0$). On the other hand, S does not do very poorly if for a honeypot ($\theta = 1$), she fails to deceive R, who simply withdraws[12] ($a = 1$).

6.5.5.2 Robustness of the Equilibrium Strategies

Finally, we investigate the performance of these strategies against suboptimal strategies of the other player. Figure 6.12 shows the expected utility for S when R acts optimally in blue, and the expected utility for S when R occasionally acts sub-optimally in red. In almost all cases, S earns higher expected utility if R plays sub-optimally than if R plays optimally.

[12]For example, consider $p = 0.1$. If R has access to a low-quality detector, then $p = 0.1$ is within the Zero-Heavy regime. Therefore, R ignores e and always chooses $a = 0$. This is a "reckless" strategy that is highly damaging to S. On the other hand, if R has access to a high-quality detector, then $p = 0.1$ is within the Middle regime. In that case, R chooses a based on e. This "less reckless" strategy actually improves S's expected utility, because R chooses $a = 0$ less often.

It can also be shown that R's strategy is robust against a suboptimal S. In fact, R's equilibrium utility remains exactly the same if S plays sub-optimally.

Corollary 6.21 *For all $\sigma^S \in \Gamma^S$, the a priori expected utility of R for a fixed σ^{R*} does not vary with σ^S.*

Proof See Sect. 6.8.6.

Corollary 6.21 states that R's equilibrium utility is not affected at all by deviations in the strategy of S. This is a result of the structure of the partially separating equilibrium. It suggests that R's equilibrium strategy performs well even if S is not fully rational.

6.6 Discussion of Results

We have proposed a holistic and quantitative model of detectable deception called signaling games with evidence. Equilibrium results include a regime in which the receiver should choose whether to trust the sender based on the detector evidence (i.e., the Middle regime), and regimes in which the receiver should ignore the message and evidence and guess the private information using only the prior probabilities (the Zero-Dominant, Zero-Heavy, One-Heavy, and One-Dominant regimes). For the sender, our results indicate that it is optimal to partially reveal the private information in the former regime. The analytical bounds that we have obtained on these regimes can be used to implement policies online, since they do not require iterative numerical computation.

We have also presented several contributions that are relevant for mechanism design. For instance, the mechanism designer can maximize the receiver utility by choosing an EER detector. Practically, designing the detector involves setting a threshold within a continuous space in order to classify an observation as an "Alarm" or "No Alarm." For an EER detector, the designer chooses a threshold that obtains equal false-positive and false-negative error rates.

At the same time, we have shown that aggressive detectors induce a truth-telling convention, while conservative detectors of the same quality induce a falsification convention. This is important if truthful communication is considered to be a design objective in itself. One area in which this applies is trust management. In both human and autonomous behavior, an agent is trustworthy if it is open and honest, if "its capabilities and limitations [are] made clear; its style and content of interactions [do] not misleadingly suggest that it is competent where it is really not" [143]. Well-designed detectors can incentivize such truthful revelation of private information.

Additionally, even deceivers occasionally prefer to reveal some true information. Numerical results have shown that the deceiver (the sender) sometimes receives a higher utility for a high-quality detector than for a low-quality detector. This result suggests that cybersecurity techniques that use defensive deception should not always

strive to eliminate leakage. Sometimes, revealing cues to deception serves as a deterrent. Finally, the strategies of the sender and receiver are robust to nonequilibrium actions by the other player. This emerges from the strong misalignment between their objectives.

6.7 Related Work

Economics literature includes several classic contributions to signaling games. Crawford and Sobel's seminal paper is the foundation of cheap-talk signaling games [38]. In this paper, a sender can just as easily misrepresent her private information as truthfully represent it. From the opposite vantage point, models from Milgrom [144], Grossman [145], and Grossman and Heart [146] study games of verifiable disclosure. In these models, a sender can choose to remain silent or to disclose information. But if the sender chooses to disclose information, then she must do so truthfully.

One way of unifying cheap-talk games and games of verifiable disclosure is to assign an explicit cost to misrepresenting the truth [39]. Cheap-talk games are a special case of this model in which the cost of lying zero, and games of verifiable disclosure are a special case in which the cost is infinite. The model presented in this chapter also bridges cheap-talk games and games of verifiable disclosure. But we do this using detection rather than lying cost. Cheap-talk games are a special case in which the detector gives alarms randomly, and games of verifiable disclosure are a special case in which the detector is perfect.

Within the literature on network security, our model is related to the *honeypot selection game with probing* studied in [96]. In this work, attackers use probes to obtain probabilistic evidence about the type of the system. In this work, evidence depends on the type, but in our work, evidence depends on both the type and the message that the sender transmits to the receiver.[13]

Finally, our model is related to information-theoretic approaches to strategic communication, such as the one in [147]. This approach combines viewpoints from economics and information-theoretic transmitter/receiver design. This approach differs from our own mainly by requiring S to optimize her strategy before she observes her private information. This makes the problem a Stackelberg game rather than a signaling game. In addition, the receiver side-channel information in this model, although similar to our concept of evidence, is correlated only with the private information, rather than with the divergence between the message and the private information as in our model.

[13] In other words, the detector in our game emits evidence when the message does not truthfully represent the type—that is, when the sender is lying. This is based on the idea that liars often give off cues of deceptive behavior.

6.8 Derivations

6.8.1 Separating PBNE

Two separating PBNE are possible. In the first both types are honest: $\sigma^S(0 \mid 0) = \sigma^S(1 \mid 1) = 1$. Then R believes they are telling the truth with certainty, regardless of the detector evidence. R trusts S and plays $a = m$. But both types of S would prefer to induce the action that the other type induces. Therefore, they deviate from the above strategies, and truth-telling is not an equilibrium.

The other type of separating PBNE is for both types of S to try to deceive R. Then R believes with certainty that they are both lying: $\mu^R(0 \mid 1, e) = 1$ and $\mu^R(1 \mid 0, e) = 1$. She plays $a \neq m$. Again, both types of S would prefer the a that the other type induces. Therefore, each type of S switches to the strategy of the other type, and complete dishonesty is not an equilibrium. In summary, no separating PBNE exist.

6.8.2 Optimal Actions of R in Pooling PBNE

Consider the case in which both types of S send $m = 0$. On the equilibrium path, Eq. (6.6) yields $\mu^R(0 \mid 0, 0) = (1 - \alpha)p(0)/((1 - \alpha)p(0) + (1 - \beta)p(1))$ and $\mu^R(0 \mid 0, 1) = \alpha p(0)/(\alpha p(0) + \beta p(1))$, while off the equilibrium path, the beliefs can be set arbitrarily. From Eq. (6.3), R chooses action $a = 0$ (e.g., R believes the signal of S) when evidence $e = 0$ and $p(0) \geq \Delta_1^R(1 - \beta)/(\Delta_0^R(1 - \alpha) + \Delta_1^R(1 - \beta))$, or when evidence $e = 1$ and $p(0) \geq \Delta_1^R\beta/(\Delta_0^R\alpha + \Delta_1^R\beta)$.

Next, consider the case in which both types of S send $m = 1$. Equation (6.6) yields $\mu^R(0 \mid 1, 0) = (1 - \beta)p(0)/((1 - \beta)p(0) + (1 - \alpha)p(1))$ and $\mu^R(0 \mid 1, 1) = \beta p(0)/(\beta p(0) + \alpha p(1))$, which leads R to choose action $a = 1$ (e.g., to believe the signal of S) if $e = 0$ and $p(0) \leq \Delta_1^R(1 - \alpha)/(\Delta_0^R(1 - \beta) + \Delta_1^R(1 - \alpha))$, or if $e = 1$ and $p(0) \leq \Delta_1^R\alpha/(\Delta_0^R\beta + \Delta_1^R\alpha)$. The order of these probabilities depends on whether $\beta > 1 - \alpha$.

6.8.3 Optimal Actions of S in Pooling PBNE

Let S pool on a message m. Consider the case that $\sigma^{R*}(1 \mid m, 0) = \sigma^{R*}(1 \mid m, 1)$, and let $a^* \triangleq \sigma^{R*}(1 \mid m, 0) = \sigma^{R*}(1 \mid m, 1)$. Then S of type $\theta = 1 - a^*$ always successfully deceives R. Clearly, that type of S does not have an incentive to deviate. But type S of type $\theta = a^*$ never deceives R. We must set the off-equilibrium beliefs such that S of type $\theta = a^*$ also would not deceive R if she were to deviate to the other message. This is the case if $\forall e \in \mathbb{EV}$, $\mu^R(a^* \mid 1 - m, e) \geq \Delta_{1-a^*}^R/(\Delta_{1-a^*}^R + \Delta_{a^*}^R)$. In that case, both types of S meet their optimality conditions, and we have a pooling PBNE.

But consider the case if $\sigma^{R*}(1 \mid m, 0) = 1 - \sigma^{R*}(1 \mid m, 1)$, (i.e., if R's response depends on evidence e). On the equilibrium path, S of type m receives utility

$$u^S(m, m, m)(1 - \alpha) + u^S(m, m, 1 - m)\alpha, \tag{6.10}$$

and S of type $1 - m$ receives utility

$$u^S(1 - m, m, 1 - m)\beta + u^S(1 - m, m, m)(1 - \beta). \tag{6.11}$$

Now we consider R's possible response to messages *off* the equilibrium path.

First, there cannot be a PBNE if R were to play the same action with both $e = 0$ and $e = 1$ off the equilibrium path. In that case, one of the S types could guarantee deception by deviating to message $1 - m$. Second, there cannot be a PBNE if R were to play action $a = m$ in response to message $1 - m$ with evidence 0 but action $a = 1 - m$ in response to message $1 - m$ with evidence 1. It can be shown that both types of S would deviate. The third possibility is that R plays action $a = 1 - m$ in response to message $1 - m$ if $e = 0$ but action $a = m$ in response to message $1 - m$ if $e = 1$. In that case, for deviating to message $1 - m$, S of type m would receive utility

$$u^S(m, 1 - m, m)\beta + u^S(m, 1 - m, 1 - m)(1 - \beta), \tag{6.12}$$

and S of type $1 - m$ would receive utility

$$u^S(1 - m, 1 - m, 1 - m)(1 - \alpha) + u^S(1 - m, 1 - m, m)\alpha. \tag{6.13}$$

Combining Eqs. (6.10)–(6.12), S of type m has an incentive to deviate if $\beta < 1 - \alpha$. On the other hand, combining Eqs. (6.11)–(6.13), S of type $1 - m$ has an incentive to deviate if $\beta > 1 - \alpha$. Therefore, if $\beta \neq 1 - \alpha$, one type of S always has an incentive to deviate, and a pooling PBNE is not supported.

6.8.4 *Partially Separating Equilibria*

For brevity, define the notation $q \triangleq \sigma^{S*}(1 \mid 0)$, $r \triangleq \sigma^{S*}(1 \mid 1)$, $w \triangleq \sigma^{R*}(1 \mid 0, 0)$, $x \triangleq \sigma^{R*}(1 \mid 0, 1)$, $y \triangleq \sigma^{R*}(1 \mid 1, 0)$, $z \triangleq \sigma^{R*}(1 \mid 1, 1)$, and $K \triangleq \Delta_1^R/(\Delta_0^R + \Delta_1^R)$.

We prove Theorem 6.14. First, assume the receiver's pure strategies $x = 1$ and $z = 0$. Second, R must choose w and y to make both types of S indifferent. This requires

$$\begin{bmatrix} \bar{\alpha} & -\bar{\beta} \\ \bar{\beta} & -\bar{\alpha} \end{bmatrix} \begin{bmatrix} w \\ y \end{bmatrix} = \begin{bmatrix} -\alpha \\ -\beta \end{bmatrix},$$

where $w, y \in [0, 1]$. Valid solutions require $\beta \leq 1 - \alpha$.

Third, S must choose q and r to make R indifferent for $(m, e) = (0, 0)$ and $(m, e) = (1, 0)$, which are the pairs that pertain to the strategies w and y. S must satisfy

$$\begin{bmatrix} -\bar{\alpha}\bar{p}\bar{K} & \bar{\beta}pK \\ -\bar{\beta}\bar{p}\bar{K} & \bar{\alpha}pK \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix} = \begin{bmatrix} -\bar{\alpha}\bar{p}\bar{K} + \bar{\beta}pK \\ 0 \end{bmatrix}.$$

Valid solutions require p to be within the Middle regime for $\beta \leq 1 - \alpha$.

Fourth, we must verify that the pure strategies $x = 1$ and $z = 0$ are optimal. This requires

$$\frac{\alpha\bar{r}p}{\alpha\bar{r}p + \beta\bar{q}\bar{p}} \leq \bar{K} \leq \frac{\beta rp}{\beta rp + \alpha q\bar{p}}.$$

It can be shown that, after substitution for q and r, this always holds. Fifth, the beliefs must be set everywhere according to Bayes' Law. We have proved Theorem 6.14.

Now we prove Theorem 6.15. First, assume the receiver's pure strategies $w = 0$ and $y = 1$. Second, R must choose x and z to make both types of S indifferent. This requires

$$\begin{bmatrix} \alpha & -\beta \\ \beta & -\alpha \end{bmatrix} \begin{bmatrix} x \\ z \end{bmatrix} = \begin{bmatrix} \bar{\beta} \\ \bar{\alpha} \end{bmatrix},$$

where $x, y \in [0, 1]$. Valid solutions require $\beta \geq 1 - \alpha$.

Third, S must choose q and r to make R indifferent for $(m, e) = (0, 1)$ and $(m, e) = (1, 1)$, which are the pairs that pertain to the strategies x and z. S must satisfy

$$\begin{bmatrix} -\alpha\bar{p}\bar{K} & \beta pK \\ -\beta\bar{p}\bar{K} & \alpha pK \end{bmatrix} \begin{bmatrix} q \\ r \end{bmatrix} = \begin{bmatrix} -\alpha\bar{p}\bar{K} + \beta pK \\ 0 \end{bmatrix}.$$

Valid solutions require p to be within the Middle regime for $\beta \geq 1 - \alpha$.

Fourth, we must verify that the pure strategies $w = 0$ and $y = 1$ are optimal. This requires

$$\frac{\alpha\bar{r}p}{\alpha\bar{r}p + \beta\bar{q}\bar{p}} \leq \bar{K} \leq \frac{\beta rp}{\beta rp + \alpha q\bar{p}}.$$

It can be shown that, after substitution for q and r, this always holds. Fifth, the beliefs must be set everywhere according to Bayes' Law. We have proved Theorem 6.15.

6.8.5 *Truth-Induction Proof*

We prove the theorem in two steps: first for the Middle regime and second for the Zero-Heavy and One-Heavy regimes.

For conservative detectors in the Middle regime, substituting the equations of Theorem 6.14 into Eq. (6.9) gives

$$\tau\left(J, G, p\right) = \frac{\bar{\alpha}\bar{\beta} - \bar{\beta}^2}{\bar{\alpha}^2 - \bar{\beta}^2} = \frac{1}{2}\left(1 - \frac{J}{1-G}\right) \leq \frac{1}{2}.$$

For aggressive detectors in the Middle regime, substituting the equations of Theorem 6.15 into Eq. (6.9) gives

$$\tau\left(J, G, p\right) = \frac{\beta^2 - \alpha\beta}{\beta^2 - \alpha^2} = \frac{1}{2}\left(1 + \frac{J}{1+G}\right) \geq \frac{1}{2}.$$

This proves the theorem for the Middle regime.

Now we prove the theorem for the Zero-Heavy and One-Heavy regimes. Since $\Delta_0^R = \Delta_1^R$, all of the Zero-Heavy regime has $p(1) \leq 1/2$, and all of the One-Heavy regime has $p(1) \geq 1/2$. For conservative detectors in the Zero-Heavy regime, both types of S transmit $m = 1$. S of type $\theta = 0$ are lying, while type $\theta = 1$ are telling the truth. Since $p(1) \leq 1/2$, we have $\tau \leq 1/2$. Similarly, in the One-Heavy regime, both types of S transmit $m = 0$. S of type $\theta = 0$ are telling the truth, while S of type $\theta = 1$ are lying. Since $p(1) \geq 1/2$, we have $\tau \leq 1/2$. On the other hand, for aggressive detectors, both types of S transmit $m = 0$ in the Zero-Heavy regime and $m = 1$ in the One-Heavy regime. This yields $\tau \geq 1/2$ in both cases. This proves the theorem for the Zero-Heavy and One-Heavy regimes.

6.8.6 *Robustness Proof*

Corollary 6.21 is obvious in the pooling regimes. In those regimes, $\sigma^{R*}(1 \mid m, e)$ has the same value for all $m \in \mathbb{M}$ and $e \in \mathbb{EV}$, so if S plays the message off the equilibrium path, then there is no change in R's action. In the mixed-strategy regimes, using the expressions for σ^{R*} from Theorems 6.14–6.15, it can be shown that, $\forall \theta \in \Theta$, $m \in \mathbb{M}$, $a \in \mathbb{A}$, $\sum_{e \in \mathbb{EV}} \lambda\left(e \mid \theta, m\right) \sigma^{R*}\left(a \mid m, 0\right) = \sum_{e \in \mathbb{EV}} \lambda\left(e \mid \theta, 1-m\right) \sigma^{R*}\left(a \mid 1-m, 0\right)$. In other words, for both types of S, choosing either message results in the same probability that R plays each actions. Since u^R does not depend on m, both messages result in the same utility for R.

6.9 Notes

Signaling games with evidence are related to *hypothesis testing*, *inspection games*, and *trust management*. Hypothesis testing evaluates the truthfulness of claims based on probabilistic evidence [148]. The evidence is emitted according to different distributions for the hypothesis under test and for the null hypothesis. Inspection games embed a hypothesis testing problem inside of a two-player game [149]. An *inspector* designs an inspection technique in order to motivate an *inspectee* to follow a rule or

regulation. The inspector chooses a probability with which to execute an inspection, and chooses whether to trust the inspectee based on the result. This chapter adds the concept of signaling to the framework of inspection games. Our model of deception can also be seen as a dual to models for trust management [46], which quantify technical and behavioral influences on the transparency and reliability of agents in distributed systems.

Chapter 7
Attacker Engagement

Advanced persistent threats (APTs) are multistage attacks that make use of social engineering and deception to give adversaries insider access to networked systems. Against APTs, active defense technologies create and exploit information asymmetry for defenders. If these active defenses are also dynamic, then we have the species of deception that Chap. 4 calls *attacker engagement*.

In the present chapter, we study a scenario in which a powerful defender uses dynamic honeynets for active defense in order to observe an attacker who has penetrated the network. Rather than immediately eject the attacker, the defender may elect to gather information. An undiscounted, infinite-horizon Markov decision process on continuous state space models the defender's problem. We find a threshold of information that the defender should gather about the attacker before ejecting him. Then we study the robustness of this policy using a Stackelberg game and simulate the policy for a conceptual network.

7.1 Introduction to Attacker Engagement

Dynamic honeynets and *virtual attack surfaces* are emerging techniques which both investigate attackers and manipulate their beliefs. They create false network views in order to lure the attacker into a designated part of a network where he can be contained and observed within a controlled environment [150]. Figure 7.1 gives a conceptual example of a honeynet placed within a process control network in critical infrastructure. A wired backbone connects wireless routers that serve sensors, actuators, controllers, and access points. A dynamic honeynet emulates a set of sensors and controllers and records attacker activities. Engaging with an attacker in order to gather information allows defenders to update their threat models and develop more effective defenses. Based on the taxonomy introduced in Chap. 4, multistage deception such as that carried out by this dynamic honeynet is called *attacker engagement*.

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_7

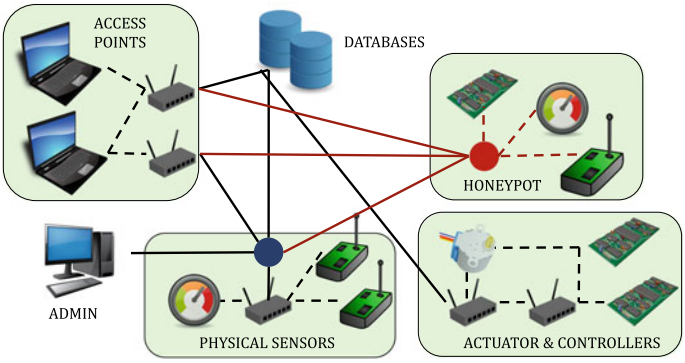


Fig. 7.1 A honeynet in a process control network. Dashed lines represent wireless connections. At the top right, a honeynet records activity in order to learn about attackers

7.1.1 Timing in Attacker Engagement

This chapter considers this seldom-studied case of a powerful defender who observes multiple attacker movements within a network. Sustained engagement with an attacker comes at the risk of added exposure. The situation gives rise to an interesting trade-off between information gathering and short-term security. How long should administrators allow an attacker to remain in a honeypot before ejecting the attacker? How long should they attempt to lure an attacker from an operational system to a honeypot? The theoretical framework in this chapter abstracts away from network topology or protocol in order to focus exclusively on these questions of timing in attacker engagement.

Section 7.2 introduces an undiscounted, infinite-horizon Markov decision process (MDP) on a continuous state space to model attacker movement constrained by a defender who can eject the attacker from the network at any time, or allow him to remain in the network in order to gather information. Section 7.3 analytically obtains the value function and optimal policy for the defender, and verifies these numerically. These results obtain closed-form conditions under which it is optimal to retain an attacker in the network. To test the robustness of the optimal policy, Sect. 7.4 develops a zero-sum, Stackelberg game model in which the attacker leads by choosing a parameter of the game. We obtain a worst-case bound on the defender's utility. Section 7.5 uses simulations to illustrate the optimal policy for a conceptual network.

Table 7.1 Notation for this chapter

Notation	Meaning
$\langle \mathbb{X} = \mathbb{L} \times \mathbb{S}, \mathbb{A}, u, q \rangle$	State space, actions, reward, transition kernel
$\mathbb{L} = [0, L^0]$	Space of residual utility
$\mathbb{S} = \{H, N, E\}$	System types (honeypot, normal, exited the network)
(L^i, S^i), $i \in 0, 1, 2, \ldots$	Residual utility and system type at stage i
A and D	Attacker and defender
$T_D = \{T_D^0, T_D^1, T_D^2, \ldots\}$	Times that D plans to wait at each stage
$T_A = \{T_A^0, T_A^1, T_A^2, \ldots\}$	Times that A plans to wait at each stage
C_N and C_H	Costs that D occurs while A is in N and H
$v \in \mathbb{R}_{++}$	Utility per unit time that D gains while A is in H
$u(L^i, S^i, T_D^i \mid T_A^i)$	One-stage reward for D
$q\left(L^{i+1}, S^{i+1}, T_D^i, L^i, S^i \mid T_A^i\right)$	Transition kernel
$p \in [0, 1]$	Fraction of normal systems in the network
$T_D^i = \sigma(L^i, S^i)$	D's strategy for residual utility L^i and system S^i
$J_\sigma^i(L^i, S^i)$	Expected infinite-horizon reward for D at stage i
$J^i\left(L^i, S^i\right)$	Expected reward for optimal policy σ^*
δ, δ_1^D, λ_N^D, χ_H^D	Value function meta-parameters
ω	Residual utility threshold
$k[\omega]$, $f^D(L^i)$	Component functions of the value function
$\bar{J}\left(\bar{T}_A\right)$	Expected utility to D over initial system types
$\bar{T}_A^*$	Worst-case $\bar{T}_A$ for D

7.2 Problem Formulation

A discrete-time, continuous state MDP can be summarized by the tuple $\langle \mathbb{X}, \mathbb{A}, u, q \rangle$, where $\mathbb{X}$ is the continuous state space, $\mathbb{A}$ is the set of actions, $u : \mathbb{X} \times \mathbb{A} \to \mathbb{R}$ is the reward function, and $q : \mathbb{X} \times \mathbb{A} \times \mathbb{X} \to \mathbb{R}_+$ is the transition kernel. In this section, we describe each of the elements of $\langle \mathbb{X}, \mathbb{A}, u, q \rangle$. Table 7.1 summarizes the notation for this chapter.

7.2.1 State Space $\mathbb{X}$

An attacker A moves throughout a network containing two types of systems S: honeypots H and normal systems N. At any time, a network defender D can eject A from the network. E denotes having left the network. Together, we have $S \in \mathbb{S} \triangleq \{H, N, E\}$.

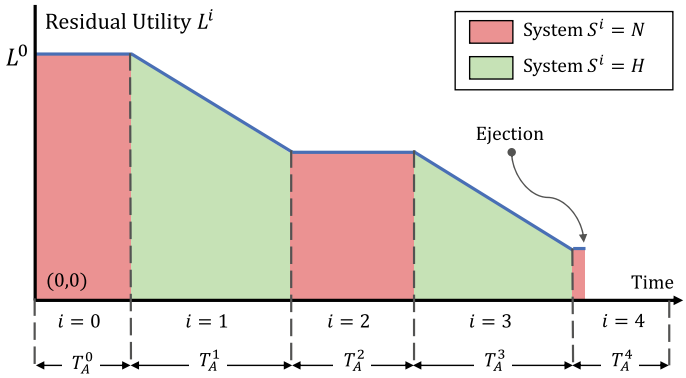


Fig. 7.2 A moves throughout a network between honeypots H and normal systems N. D can earn a total of L^0 utility for investigating A. When A is in a honeypot, D learns and the residual utility for future investigation decreases. Near $L^i = 0$, the risk of exposure outweighs the benefit of surveillance, and D ejects A at stage $i = 4$ (in this example)

Let $i \in 0, 1, 2, \ldots$ denote the discrete *stage* of the game, i.e., i indicates the order of the systems visited. D observes the types S^i of the systems that A visits. The attacker, on the other hand, does not know the system types.

We assume that there is a maximum amount of information that D can learn from investigating A. Let L^0 denote the corresponding utility that D receives for this information. At stage $i \in 0, 1, 2, \ldots$, let $L^i \in \mathbb{L} \triangleq [0, L^0]$ denote the *residual utility*[1] available to D for investigating A. For instance, at $i = 5$, D may have recorded the attacker's time of infiltration, malware type, and operating system, but not yet any privilege escalation attempts, which could reveal the attacker's objective. In that case, D may estimate that $L^5 \approx 0.6L^0$, i.e., D has learned approximately 60% of all possible information[2] about A.

D should use L^i together with S^i to form his policy. For instance, with $L^5 \approx 0.6L^0$, D may allow A to remain in a honeypot $S^5 = H$. But after observing a privilege escalation attempt, with $L^6 \approx 0.8L^0$, D may eject A from $S^6 = H$, since there is little more to be learned about him. Therefore, L^i and S^i are both states. The full state space is $\mathbb{X} = \mathbb{L} \times \mathbb{S}$. Figure 7.2 summarizes the interaction.

[1] Quantification of control- and game-theoretic utility parameters (such as L^0) is never exact, and requires substantial effort. In this scenario, D must first judge what type of attacker he is facing. If it is a simple, brute-force botnet scan, L^0 will be low, while if it is a complex, targeted attack coming from a state actor, L^0 will be high.

[2] Here D must specify the pieces of information that he hopes to learn about A, as well as their relative values.

7.2.2 One-Stage Actions $\mathbb{A}$

Let $\aleph_0$ denote the cardinality of the set of natural numbers and $\mathbb{R}_+$ denote the set of nonnegative real numbers. Then define $T_D = \{T_D^0, T_D^1, T_D^2, \ldots\} \in \mathbb{R}_+^{\aleph_0}$ such that T_D^i denotes the time that D plans to wait at stage i before ejecting A from the network. The single-stage action of D is to choose $T_D^i \in \mathbb{A} = \mathbb{R}_+$.

7.2.3 Reward Function u

To formulate the reward, we also need to define $T_A = \{T_A^0, T_A^1, T_A^2, \ldots\} \in \mathbb{R}_+^{\aleph_0}$. For each $i \in 0, 1, 2, \ldots, T_A^i$ denotes the duration of time that A plans to wait at stage i before changing to a new system.

Let $C_N < 0$ denote the average cost per unit time that D incurs while A resides in normal systems.[3] This cost may be estimated by a sum of the costs $\phi_j^m < 0$ per unit time of each known vulnerability $j \in 1, 2, \ldots, J-1$ on each of the systems $m \in 1, 2, \ldots, M$ in the network, weighted by the likelihoods $\rho_j^m \in [0, 1]$ that A exploits the vulnerability. This data can be obtained from the National Vulnerabilities Database [151]. Of course, D may not be aware of some system vulnerabilities. Let ϕ_J^m denote an estimate of the damage that could be caused by an unknown vulnerability J on each system $m \in 1, 2, \ldots, M$, and let ρ_J^m denote a heuristic likelihood that A can exploit the vulnerability.[4]

$$C_N = \frac{1}{M}\sum_{m=1}^{M} C_N^m = \frac{1}{M}\sum_{m=1}^{M}\sum_{j=1}^{J} \rho_j^m \phi_j^m.$$

Let $C_H \leq 0$ denote a cost that D pays to maintain A in a honeypot. This cost could represent the expense of hiring personnel to monitor the honeypot, the expense of redeployment, or the loss of informational advantage from A reconnoitering the honeypot. Let $\mathbb{R}_{++}$ denote the set of strictly positive real numbers. Finally, let $v \in \mathbb{R}_{++}$ denote the utility per unit time that D gains from learning about A while he is in honeypots. We assume that $v > -C_H$, i.e., that the benefit per unit time from observing A in a honeypot exceeds the cost.

Define the function $u : \mathbb{L} \times \mathbb{S} \times \mathbb{R}_+ \to \mathbb{R}$ such that $u(L^i, S^i, T_D^i \mid T_A^i)$ gives the one-stage reward to D if the residual utility is L^i, A is in system S^i, A waits for T_A^i before moving, and D waits for T_D^i before ejecting A. Let $T^i \triangleq \min(T_A^i, T_D^i)$ denote the time for which A remains at system S^i before moving or being ejected. Also, let $\mathbf{1}\{P\}$ be the indicator function which returns 1 if it the statement P is true. We have $u(L^i, S^i, T_D^i \mid T_A^i) =$

[3] Future work can consider different costs for each individual system in a structured network.

[4] One alternate approach that can be used to quantify the impact of unknown zero-day attacks is k-zero day safety [152].

$$\mathbf{1}\{S = N\}C_N T^i + \mathbf{1}\{S = H\}\left(\min\left(T^i v, L^i\right) + C_H T^i\right).$$

7.2.4 Transition Kernel q

Let $\mathbb{R}_+$ denote the set of nonnegative real numbers. For stage $i \in 0, 1, 2, \ldots,$ and given attacker and defender move times T_A^i and T_D^i, respectively, define the transition kernel $q : \mathbb{L} \times \mathbb{S} \times \mathbb{R}_+ \times \mathbb{L} \times \mathbb{S} \to \mathbb{R}_+$ such that, for all residual utilities $L^i \in \mathbb{L}$ and system types $S^i \in \mathbb{S}$

$$\int_{L^{i+1}\in\mathbb{L}} \int_{S^{i+1}\in\mathbb{S}} q\left(L^{i+1}, S^{i+1}, T_D^i, L^i, S^i \mid T_A^i\right) = 1,$$

where L^{i+1} and S^{i+1} denote the residual utility and system type, respectively, at the next stage.

Let $p \in [0, 1]$ denote the fraction of normal systems in the network.[5] For a real number y, let $\delta(y)$ be the Dirac delta function. For brevity, let $\Phi(L^i, T) \triangleq \max\{L^i - vT, 0\}$. If $T_A^i > T_D^i$, then D ejects A from the system, and we have $q(L^{i+1}, S^{i+1}, T_D^i, L^i, S^i \mid T_A^i) =$

$$\begin{aligned}
&\mathbf{1}\left\{S^i = E \cap S^{i+1} = E\right\}\delta\left(L^{i+1} - L^i\right) + \\
&\qquad \mathbf{1}\left\{S^i = N \cap S^{i+1} = E\right\}\delta\left(L^{i+1} - L^i\right) + \\
&\qquad\qquad \mathbf{1}\left\{S^i = H \cap S^{i+1} = E\right\}\delta\left(L^{i+1} - \Phi(L^i, T_D^i)\right). \qquad (7.1)
\end{aligned}$$

If $T_A^i \le T_D^i$, then A changes systems, and we have $q\left(L^{i+1}, S^{i+1}, T_D^i, L^i, S^i \mid T_A^i\right) =$

$$\begin{aligned}
&p\mathbf{1}\left\{S^i = N \cap S^{i+1} = N\right\}\delta\left(L^{i+1} - L^i\right) + \\
&\qquad (1-p)\,\mathbf{1}\left\{S^i = N \cap S^{i+1} = H\right\}\delta\left(L^{i+1} - L^i\right) + \\
&\qquad p\mathbf{1}\left\{S^i = H \cap S^{i+1} = N\right\}\delta\left(L^{i+1} - \Phi(L^i, T_A^i)\right) + \\
&\qquad\qquad (1-p)\,\mathbf{1}\left\{S^i = H \cap S^{i+1} = H\right\}\delta\left(L^{i+1} - \Phi(L^i, T_A^i)\right). \qquad (7.2)
\end{aligned}$$

The equations can be understood by considering an example. If $T_A^i \le T_D^i$ and A is currently in a honeypot, then Eq. (7.2) shows that the remaining utility will be $\delta\left(L^{i+1} - \Phi(L^i, T_A^i)\right)$. There is p probability that the next system is N, and $(1-p)$ that it is H.

[5] Again, in a formal network, the kernel will differ among different honeypots and different normal systems. The fraction p is an approximation which is exact for a fully-connected network.

7.2.5 *Infinite-Horizon, Undiscounted Reward*

For stage $i \in 0, 1, 2, \ldots,$ define the stationary deterministic feedback policy $\sigma : \mathbb{L} \times \mathbb{S} \rightarrow \mathbb{R}_+$ such that $T_D^i = \sigma(L^i, S^i)$ gives the time that D waits before ejecting A if the residual utility is L^i and the system type is S^i. Let Θ denote the space of all such stationary policies. Define the expected infinite-horizon, undiscounted reward by $J_\sigma^i : \mathbb{L} \times \mathbb{S} \rightarrow \mathbb{R}$ such that $J_\sigma^i(L^i, S^i)$ gives the expected reward from stage i onward for using the policy σ when the residual utility is L^i and the type of the system is S^i. $J_\sigma^i(L^i, S^i)$ also depends on the attacker's choice of T_A^k, but for a given T_A^k, it is expressed by

$$J_\sigma^i\left(L^i, S^i\right) = \mathbb{E}\left\{\sum_{k=i}^{\infty} u\left(L^k, S^k, \sigma\left(L^k, S^k\right) \mid T_A^k\right)\right\},$$

where the states transition according to Eqs. (7.1)–(7.2). Given an initial system type $S^0 \in \{H, N\}$, the overall problem for D is to find σ^* such that

$$\sigma^* \in \arg\max_{\sigma \in \Theta} J_\sigma^0\left(L^0, S^0\right).$$

The undiscounted utility function demands Proposition 7.1.

Proposition 7.1 *$J_{\sigma^*}^i(L^i, S^i)$ is finite.*

Proof See Appendix 7.8.1.

It is also convenient to define the *value function* as the reward for the optimal policy

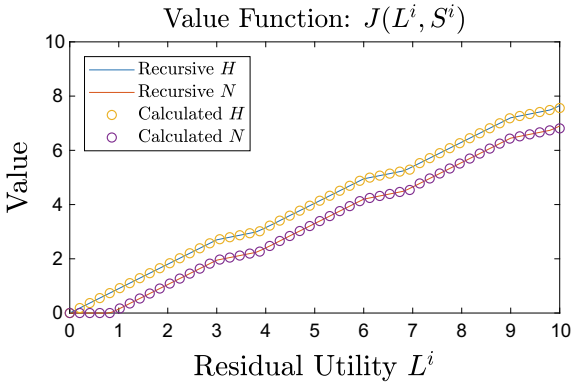
$$J^i\left(L^i, S^i\right) \triangleq J_{\sigma^*}^i\left(L^i, S^i\right) = \max_{\sigma \in \Theta} J_\sigma^i\left(L^i, S^i\right).$$

The Bellman principle [153] implies that for an optimal stationary policy σ^*, and for $i \in 0, 1, 2, \ldots, \sigma^*\left(L^i, S^i\right) \in$

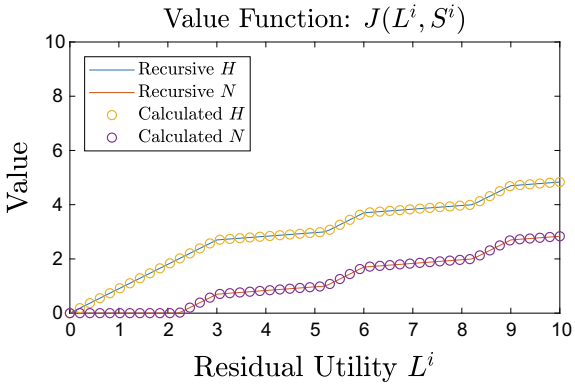
$$\arg\max_{T_D^i \in \mathbb{R}_+} u\left(L^i, S^i, T_D^i \mid T_A^i\right) + \int_{L^{i+1} \in \mathbb{L}} \int_{S^{i+1} \in \mathbb{S}} J^{i+1}\left(L^{i+1}, S^{i+1}\right) q(L^{i+1}, S^{i+1}, T_D^i, L^i, S^i \mid T_A^i).$$

7.3 Analysis and Results

In this section, we solve for the value function and optimal policy. We start by obtaining the optimal policy in honeypots, and reducing the space of candidates for an optimal policy in normal systems. Then we present the value function and optimal policy separately, although they are derived simultaneously.



(a) $p = 0.60$, $\omega \approx 0.83$, $\delta = 3.0$, $L^0 = 10$



(b) $p = 0.85$, $\omega \approx 2.2$, $\delta = 3.0$, $L^0 = 10$

Fig. 7.3 Value functions with $p = 0.60$ and $p = 0.85$. The top and bottom curves depict $J(L^i, H)$ and $J(L^i, N)$, respectively, as a function of L^i. The circles plot the analytical $J(L^i, S)$, $S \in \{H, N\}$ from Theorem 7.9 and the solid lines verify this using an iterative numerical method

7.3.1 *Reduced Action Spaces*

Lemma 7.2 obtains the optimal waiting time for $S^i = H$.

Lemma 7.2 (Optimal Policy for $S^i = H$) *In honeypots, for any* $i \in 0, 1, 2, \ldots$ *and* $L^i \in \mathbb{L}$, *the value function is optimized by playing* $T_D^i = L^i/v$.

Proof The value of the game is maximized if A passes through only honeypots and D ejects A when the residual utility is 0. D can achieve this by playing $T_D^i = L^i/v$ if $T_A^i > L^i/v$. On the other hand, if $T_A^i \leq L^i/v$, then it is optimal for D to allow A to change systems. This is optimal because the value function at stage $i + 1$ is nonnegative, since in the worst-case D can eject A immediately if A arrives at a normal system. D can allow A to change systems by playing any $T_D^i \geq T_A^i$, although it is convenient for brevity of notation to choose $T_D^i = T_A^i$.

Lemma 7.3 narrows the optimal waiting times for $S^i = N$.

Lemma 7.3 (Reduced Action Space for $S^i = N$) *In normal systems, for any* $i \in 0, 1, 2, \ldots$ *and* $L^i \in \mathbb{L}$, *the value function is optimized by playing either* $T_D^i = 0$ *or* $T_D^i = T_A^i$.

Proof First, note that it is always suboptimal for D to eject A at a time less that T_A^i. That is, for stage $i \in 0, 1, 2, \ldots$, $J_{\tilde{\sigma}}^i(L^i, N) < J_{\hat{\sigma}}^i(L^i, N)$ for $0 = \hat{\sigma}(L^i, N) < \tilde{\sigma}(L^i, N) < T_A^i$. Second, note that D receives the same utility for ejecting A at any time greater than or equal to T_A^i, i.e., $J_{\tilde{\sigma}}^i(L^i, N) = J_{\hat{\sigma}}^i(L^i, N)$ for $T_A^i \leq \hat{\sigma}(L^i, N) \leq \tilde{\sigma}(L^i, N)$. Then either 0 or T_A^i is optimal.

Remark 7.4 summarizes Lemmas 7.2–7.3.

Remark 7.4 Lemma 7.2 obtains the unique optimal waiting time in honeypots. Lemma 7.3 reduces the candidate set of optimal waiting times in normal systems to two times: $T_D^i \in \{0, T_A^i\}$. These times are equivalent to stopping the Markov chain and allowing it to continue, respectively. Thus, Lemmas 7.2–7.3 show that the MDP is an optimal stopping problem.

7.3.2 Value Function Structure

To solve the optimal stopping problem, we must find the value function. We obtain the value function for a constant attacker action, i.e., $T_A^0 = T_A^1 = ... \triangleq \bar{T}_A$. This means that $J^i \equiv J$. Define the following notation:

$$\delta \triangleq \bar{T}_A v, \quad \delta_1^D \triangleq \bar{T}_A (v + C_H), \tag{7.3}$$

$$\lambda_N^D \triangleq \frac{-C_N}{1-p}, \quad \chi_H^D \triangleq \frac{v + C_H}{v}. \tag{7.4}$$

Note that δ and δ_1^D are in units of utility, λ_N^D is in units of utility per second, and χ_H^D is unitless.

First, $J(L^i, E) = 0$ for all $L^i \in \mathbb{L}$, because no further utility can be earned after D ejects A. Next, $J(0, S) = 0$ for both $S \in \{H, N\}$, because no positive utility can be earned in either type of system. J can now be solved backwards in L^i from $L^i = 0$ to $L^i = L^0$ using these terminal conditions. Depending on the parameters, it is possible that $\forall L^i \in \mathbb{L}$, $\sigma^*(L^i, N) = 0$ and $J(L^i, N) = 0$, i.e., D should eject A from all normal systems immediately. Lemma 7.5 describes the structure of the optimal policy outside of this case.

Lemma 7.5 (Optimal Policy Structure) *Outside of the case that* $\forall L^i \in \mathbb{L}$, $\sigma^*(L^i, N) = 0$, *there exists a residual utility* $\omega \in \mathbb{L}$ *such that:*

- for $L^i < \omega$, $\sigma^*(L^i, N) = 0$ and $J(L^i, N) = 0$,
- for $L^i > \omega$, $\sigma^*(L^i, N) = \bar{T}_A$ and $J(L^i, N) > 0$.

Proof See Appendix 7.8.2.

Remark 7.6 Typical intuition dictates that a security professional should immediately eject a detected attacker from normal systems in a network. Lemma 7.5 shows that this is indeed optimal when $\omega \geq L^0$. When $\omega < L^0$, however, it is better to allow the attacker to remain. A principal contribution of our work is finding this threshold ω.

7.3.3 Value Function Threshold

Next, for $x \in \mathbb{R}$, define

$$k[x] \triangleq \begin{cases} \lfloor x/\delta \rfloor, & \text{if } x \geq 0 \\ 0, & \text{if } x < 0 \end{cases}, \tag{7.5}$$

where $\lfloor \bullet \rfloor$ is the floor function. The floor function is required because u is nonlinear in L^i. Then Theorem 7.7 gives ω in closed form.

Theorem 7.7 (Threshold ω) *Outside of the trivial case, the threshold ω of residual utility beyond which D should eject A is given by*

$$\omega = \delta \left(k[\omega] + \frac{\lambda_N^D}{(v + C_H)(1-p)^{k[\omega]}} - \frac{1-(1-p)^{k[\omega]}}{p(1-p)^{k[\omega]}} \right),$$

where $k[\omega]$ is defined as in Eq. (7.5), and it can be shown that

$$k[\omega] = \left\lfloor \log_{1-p} \left(1 + \frac{pC_N}{(1-p)(v + C_H)} \right) \right\rfloor,$$

if the argument of the logarithm is positive. If not, then the optimal policy is for D to eject A from normal systems immediately.

Proof See Appendix 7.8.3.

Remark 7.8 gives some intuition about Theorem 7.7.

Remark 7.8 Numerical results suggest that in many cases (such as those in Fig. 7.3), $k[\omega] = 0$. In that case, we have $\omega = -\delta C_N / ((v + C_H)(1-p))$. The threshold ω increases as the cost for normal systems (C_N) increases, decreases as the rate at which utility is gained in normal systems (v) increases, and decreases as the proportion of normal systems (p) increases.

Finally, Theorem 7.9 summarizes the value function.

Theorem 7.9 (Value Function) *The value function is given by*

$$J\left(L^i, S^i\right) = \begin{cases} 0, & \text{if } S^i = E \\ f^D(L^i), & \text{if } S^i = H \\ \left\{f^D(L^i) - \bar{T}_A \lambda_N^D\right\}_+, & \text{if } S^i = N \end{cases},$$

where $\{\bullet\}_+$ *denotes* $\max\{\bullet, 0\}$, *and* $f^D : \mathbb{L} \to \mathbb{R}_+$ *is*

$$f^D\left(L^i\right) \triangleq \chi_H^D\left(L^i - \delta k[L^i]\right)(1-p)^{k[L^i]-k[L^i-\omega]} + \frac{\delta_1^D}{p}\left(1-(1-p)^{k[L^i]-k[L^i-\omega]}\right) + k[L^i-\omega]\left(\delta_1^D - p\lambda_N^D \bar{T}_A\right).$$

Proof See Appendix 7.8.2.

Remark 7.10 discusses the interpretation of Theorem 7.9.

Remark 7.10 The quantity $f^D(L^i)$ is the expected reward for future surveillance, while $\bar{T}_A \lambda_N^D$ is the expected damage that will be caused by A. In normal systems, when $L^i \leq \omega$, we have $f^D(L^i) \leq \bar{T}_A \lambda_N^D$, and the risk of damage outweighs the reward of future surveillance. Therefore, it is optimal for D to eject A, and $J(L^i, N) = 0$. On the other hand, for $L^i > \omega$, it is optimal for D to allow A to remain for $\bar{T}_A$ before moving, so $J(L^i, N) > 0$. Figure 7.3 gives examples of the value function.

7.3.4 Optimal Policy Function

Theorem 7.11 summarizes the optimal policy.

Theorem 7.11 (Defender Optimal Policy) *D achieves an optimal policy for* $S^i \in \{H, N\}$ *by playing*

$$\sigma^*\left(L^i, S^i\right) = \begin{cases} L^i/v, & \text{if } S^i = H \\ \bar{T}_A, & \text{if } S^i = N \text{ and } L^i \geq \omega \\ 0, & \text{if } S^i = N \text{ and } L^i < \omega \end{cases}.$$

Proof See Appendix 7.8.2.

7.4 Robustness Evaluation

In this section, we evaluate the robustness of the policy σ^* by allowing A to choose the worst-case $\bar{T}_A$.

7.4.1 Equilibrium Concept

Let us write $J_\sigma(L^i, S^i \mid \bar{T}_A)$ and $\sigma^*(L^i, S^i, \mid \bar{T}_A)$ to denote the dependence of the value and optimal policy, respectively, on $\bar{T}_A$. Next, define $\bar{J} : \mathbb{R}_+ \to \mathbb{R}$ such that $\bar{J}(\bar{T}_A)$ gives the expected utility to D over possible types of initial systems for playing σ^* as a function of $\bar{T}_A$. This is given by

$$\bar{J}\left(\bar{T}_A\right) = pJ\left(L^0, N \mid \bar{T}_A\right) + (1-p)\, J\left(L^0, H \mid \bar{T}_A\right). \tag{7.6}$$

Definition 7.12 formulates a *zero-sum Stackelberg equilibrium* [81] in which A chooses $\bar{T}_A$ to minimize Eq. (7.6), and D plays the optimal policy given $\bar{T}_A$ from Theorem 7.11.

Definition 7.12 (*Stackelberg Equilibrium*) A Stackelberg equilibrium (SE) of the zero-sum attacker-defender game is a strategy pair $(\bar{T}_A^*, \sigma^*)$ such that

$$\bar{T}_A^* \in \arg\min_{\bar{T}_A} \bar{J}_{\sigma^*(L^i, S^i \mid \bar{T}_A)}\left(\bar{T}_A\right),$$

and $\forall L^i \in \mathbb{L}$, $\forall S^i \in \mathbb{S}$,

$$\sigma^*\left(L^i, S^i \mid \bar{T}_A^*\right) \in \arg\max_{\sigma \in \Theta} J_\sigma\left(L^i, S^i \mid \bar{T}_A^*\right).$$

Definition 7.12 considers A as the Stackelberg game leader because our problem models an intelligent defender who reacts to the strategy of an observed attacker.

7.4.2 Equilibrium Analysis

$\bar{J}_{\sigma^*}(\bar{T}_A)$ takes two possible forms, based on the values of δ and ω. Figure 7.4 depicts $\bar{J}_{\sigma^*}(\bar{T}_A)$ for $\delta < \omega$, and Fig. 7.5 depicts $\bar{J}_{\sigma^*}(\bar{T}_A)$ for $\delta > \omega$. Note that the oscillations are not produced by numerical approximation, but rather by the nonlinear value function.[6] The worst-case $\bar{T}_A^*$ is as small as possible for $\delta < \omega$ and is large for $\delta > \omega$. Theorem 7.13 states this result formally.

Theorem 7.13 (Value as a function of $\bar{T}_A$) *For low* $\bar{T}_A$:

$$\lim_{\bar{T}_A \to 0} \bar{J}_{\sigma^*}(\bar{T}_A) = L^0\left(1 + \frac{1}{v}\left(C_H + C_N \frac{p}{1-p}\right)\right). \tag{7.7}$$

Define $\bar{T}_\omega$ *as* $\bar{T}_A$ *such that* $L^0 = \omega$. *Then for* $\bar{T}_A \geq \max\{r, \bar{T}_\omega\}$, *we have*

[6] As $\bar{T}_A$ varies, the number of systems that A can visit before $L^i < \omega$ changes in a discrete manner. This causes the oscillations.

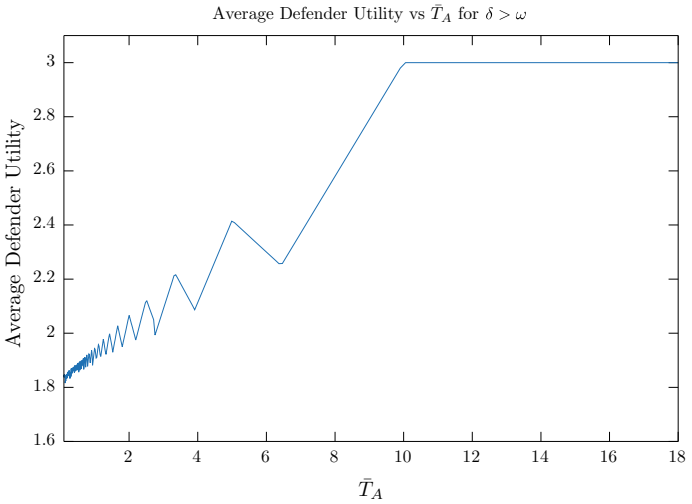


Fig. 7.4 $\bar{J}_{\sigma^*}(\bar{T}_A)$ for the case that $\delta < \omega$. Here, the worst-case value is $\bar{J}_{\sigma^*}(\bar{T}_A^*) \approx 1.8$, which occurs as $\bar{T}_A \to 0$

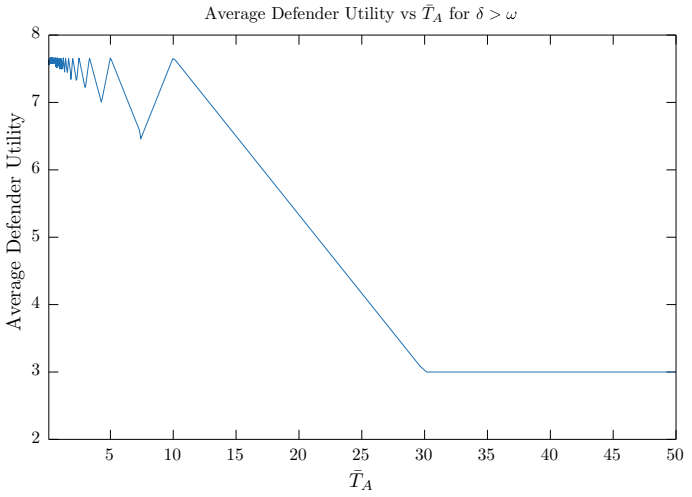


Fig. 7.5 $\bar{J}_{\sigma^*}(\bar{T}_A)$ for the case that $\delta > \omega$. Here, the worst-case value is $\bar{J}_{\sigma^*}(\bar{T}_A^*) \approx 3.0$, which occurs for $\bar{T}_A > \omega \approx 30$

$$\bar{J}_{\sigma^*}(\bar{T}_A) = L^0 (1-p) \frac{v + C_H}{v}. \tag{7.8}$$

Proof See Appendix 7.8.4.

Remarks 7.14–7.15 discuss Theorem 7.13 and Figs. 7.4 and 7.5.

Remark 7.14 The parameters of Figs. 7.4 and 7.5 differ only in C_N, which has a higher absolute value in Fig. 7.4. Since C_N only affects $\bar{J}_{\sigma^*}(\bar{T}_A)$ as $\bar{T}_A \to 0$, the plots are the same for high $\bar{T}_A$.

Remark 7.15 The connection between Figs. 7.4 and 7.5 can be visualized by translating the left sides of the curves vertically, while the right sides remain fixed. This gives network designers an intuition of how the worst-case value can be manipulated by changing the parameters of the game.

Finally, Corollary 7.16 summarizes the worst-case value.

Corollary 7.16 (Worst-Case Value) *The worst-case value* $\bar{J}_{\sigma^*}(\bar{T}_A^*)$ *is approximated by*

$$L^0 \min_{\bar{T}_A} \left\{ \left(1 + \frac{1}{v} \left(C_H + C_N \frac{p}{1-p} \right) \right), (1-p) \frac{v + C_H}{v} \right\}.$$

7.5 Simulation

In this section, we simulate a network which sustains five attacks and implements D's optimal policy σ^*. Consider the example network depicted in Fig. 7.1 in Sect. 7.1. This network has 16 production nodes, including routers, wireless access points, wired admin access, and a database. It also has sensors, actuators, and controllers, which form part of a SCADA (supervisory control and data acquisition) system. The network has 4 honeypots (in the top right of the figure), configured to appear as additional SCADA system components.

Figure 7.6 depicts a view of the network in MATLAB [154]. The red line indicates an attack path, which enters through the wireless access point at node 1, passes through the honeynet in nodes 11, 18, and 19, and enters the SCADA components in nodes 6 and 7. The transitions are realized randomly.

Figure 7.7 depicts the cumulative utility of D over time for five simulated attacks. Towards the beginning of the attacks, D gains utility. But after learning nears completion (i.e., $L^i \approx 0$), the losses C_N from normal systems dominate. The filled boxes in each trace indicate the ejection point dictated by σ^*. At these points, $L^i \leq \omega$. The ejection points are approximately at the maximum utility for traces 1, 3, and 5, and obtain a positive utility in trace 4. Trace 5 involves a long period in which $S^i = N$, and D sustains heavy losses. Since the traces are realized randomly, σ^* maximizes expected utility rather than realized utility.

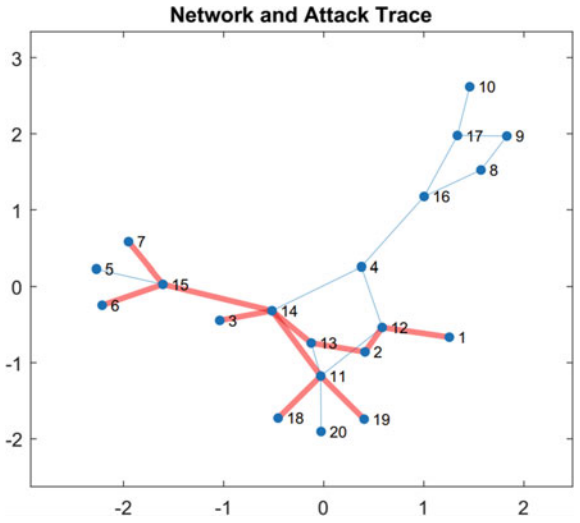


Fig. 7.6 The blue nodes and edges illustrate a 20-node network and the red highlights indicate an example attack trace

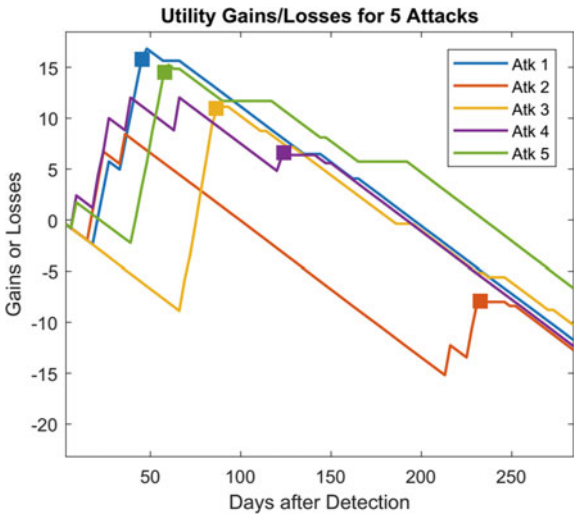


Fig. 7.7 The curves indicate the cumulative utility gains or losses for five simulated attacks. The solid squares indicate the optimal ejection time according to σ^*

7.6 Discussion of Results

This chapter aimed to assess how long an intelligent network defender that detects an attacker should observe the attacker before ejecting him. We found that the defender should keep the attacker in a honeypot as long as the information remains to be learned, and should keep him in a normal system until a threshold amount of information remains. This threshold is ω, at which the benefits of observation exactly balance the risks of information loss. Using this model, network designers can vary parameters (e.g., the number of honeypots and the rate at which they gather informa-

tion) in order to maximize the value function J. In particular, we have examined the effect of the attacker move period $\bar{T}_A$ using a Stackelberg game in which A chooses the worst-case $\bar{T}_A$. Future work can use signaling games to calculate attacker beliefs p and $1 - p$ based on defender strategies. Another direction, for distributed sensor-actuator networks, is to quantify the risk C_N of system compromise using optimal control theory.

7.7 Related Work

Game-theoretic design of honeypot deployment has been an active research area. Signaling games are used to model attacker beliefs about honeypots in [50, 140, 155]. Honeynet deployment from a network point of view is systematized in [150]. Reference [156] develops a model for lateral movements and formulates a game by which an automated defense agent protects a network asset. Durkota et al. model dynamic attacker engagement using attack graphs and a MDP [110]. Zhuang et al. study security investment and deception using a multiple round signaling game [109]. This chapter fits within the context of these papers, but focuses on questions of timing.

7.8 Derivations

7.8.1 Proof of Finite Expected Value

The maximum value of $J^i_\sigma(L^i, S^i)$ is achieved if A only visits honeypots. In this case, $J^i_\sigma(L^i, S^i) = (v + C_H)L^0/v$, so the expected utility is bounded from above. If D chooses a poor policy (for example, $\sigma(L^i, S^i) = T^i_A$ for all $L^i \in \mathbb{L}$ and $S^i \in \mathbb{S}$), then $J^i_\sigma(L^i, S^i)$ can be unbounded below. On the other hand, D can always guarantee $J^i_\sigma(L^i, S^i) = 0$ (for example, by choosing $\sigma(L^i, S^i) = 0$ for all $L^i \in \mathbb{L}$ and $S^i \in \mathbb{S}$). Therefore, the value of the *optimal* policy is bounded from below as well as from above.

7.8.2 Derivation of Value Function and Optimal Policy

For $S^i \in \{H, N\}$, the value function $J(L^i, S^i)$ is piecewise-linear in L^i. The pieces result from different discrete numbers of systems that A visits. Let $J(L^i, S^i)[a, b]$ denote $J(L^i, S^i)$ restricted to the domain $L^i \in [a, b] \subset \mathbb{R}$. First, we find $J(L^i, N)$ in terms of $J(L^i, H)$. For any nonnegative integer k, one step of the Bellman equation gives $J(L^i, N)[k\delta, (k + 1)\,\delta] =$

$$\left\{C_N \bar{T}_A + pJ\left(L^i, N\right)[k\delta, (k+1)\,\delta] + (1-p)\,J\left(L^i, H\right)[k\delta, (k+1)\,\delta]\right\}_+,$$

where $\{\bullet\}_+$ denotes $\max\{\bullet, 0\}$. D achieves this maximization by continuing the game if the expected value for continuing is positive, and ejecting A if the expected value is negative.

Rearranging terms and using Eqs. (7.3)–(7.4) gives $J(L^i, N)\,[k\delta, (k+1)\,\delta] = \left\{J\left(L^i, H\right)[k\delta, (k+1)\,\delta] - \lambda_N^D \bar{T}_A\right\}_+$. Now, we have defined ω as $L^i \in \mathbb{R}_+$ which makes the argument on the right side equal to zero. This obtains $J\left(L^i, N\right)$ $[k\delta, (k+1)\,\delta] =$

$$\begin{cases} 0, & \text{if } L^i \le \omega \\ J\left(L^i, H\right)[k\delta, (k+1)\,\delta] - \lambda_N^D \bar{T}_A, & \text{if } L^i > \omega. \end{cases}$$

Next, we find $J(L^i, H)$. First, consider $J(L^i, H)[0, \delta]$. D keeps A in the honeypot until all residual utility is depleted, and then ejects him. Thus $J(L^i, H)[0, \delta] = L^i \chi_H^D$. Next, for $k \in 1, 2, \ldots$, consider $J(L^i, H)[k\delta, (k+1)\delta]$. We have $J\left(L^i, H\right)$ $[k\delta, (k+1)\,\delta] =$

$$(v + C_H)\,\bar{T}_A + pJ\left(L^i - \delta, N\right)[(k-1)\,\delta, k\delta] + (1-p)\,J\left(L^i - \delta, H\right)[(k-1)\,\delta, k\delta].$$

A bit of algebra gives

$$J(L^i, H)[k\delta, (k+1)\,\delta] = \delta_1^D + (1-p)\,J\left(L^i - \delta, H\right)[(k-1)\,\delta, k\delta],$$

if $L^i \le \omega + \delta$, and $J(L^i, H)[k\delta, (k+1)\,\delta] = \delta_1^D + J\left(L^i - \delta, H\right)[(k-1)\,\delta, k\delta] - p\lambda_N^D \bar{T}_A$, otherwise. Solving this recursive equation for the case of $L^i \le \omega + \delta$ gives $J(L^i, H)[k\delta, (k+1)\,\delta] =$

$$\delta_1^D + \delta_1^D\,(1-p) + \ldots + \delta_1^D\,(1-p)^{k-1} + (1-p)^k\,J\left(L^i - \delta k, H\right)[0, \delta]. \quad (7.9)$$

Using initial condition $J(L, H)[0, \delta] = L\chi_H^D$ produces $f^D(L^i)$ for $L^i \le \omega$. For $L^i > \omega + \delta$, consider the integer k_1 such that $(k - k_1 - 1)\delta \le \omega < (k - k_1)\delta$. Then

$$J(L^i, H)[k\delta, (k+1)\,\delta] = k_1\left(\delta_1^D - p\lambda_N^D \bar{T}_A\right) + J\left(L^i - k_1\delta, H\right)[(k-k_1-1)\delta, (k-k_1)\delta].$$

But the last term is simply $f^D\left(L^i - k_1\delta\right)$, and $k_1 = k\left[L^i - \omega\right]$ defined in Eq. (7.5). Substituting from Eq. (7.9) gives the entire function $f^D(L^i)$, $L^i \in \mathbb{L}$.

7.8.3 Derivation of $k[\omega]$ and ω

We solve first for $k[\omega]$ and then for ω. Because of the floor function in $k[\omega]$, we have that $\omega \in [k[\omega]\delta, (k[\omega]+1)\delta)$. Then for some $\epsilon \in [0,1)$, $\omega = (k[\omega]+\epsilon)\delta$.

Note that $f^D(\omega) = \bar{T}_A \lambda_N^D$, i.e., the expected gain of surveillance is equal to the security risk at $L^i = \omega$. Therefore, we have $\bar{T}_A \lambda_N^D =$

$$\chi_H^D (\omega - \delta k[\omega]) (1-p)^{k[\omega]} + \frac{\delta_1^D}{p}\left(1-(1-p)^{k[\omega]}\right). \tag{7.10}$$

Substituting for ω,

$$\bar{T}_A \lambda_N^D - \frac{\delta_1^D}{p} = (k[\omega]+\epsilon)\delta\chi_H^D (1-p)^{k[\omega]} - \delta k[\omega]\chi_H^D (1-p)^{k[\omega]} - (1-p)^{k[\omega]}.$$

This reduces to

$$\bar{T}_A \lambda_N^D - \frac{\delta_1^D}{p} = \epsilon\delta\chi_H^D (1-p)^{k[\omega]} - (1-p)^{k[\omega]},$$

which is uniquely solved by the $k[\omega]$ in Theorem 7.7. Now solving Eq. (7.10) for ω obtains the result in Lemma 7.7.

7.8.4 Derivation of $\bar{J}_{\sigma^*}(\bar{T}_A)$

We solve the value function in two cases.

Limit as $\bar{T}_A \to 0$

As $\bar{T}_A \to 0$, ω and δ decrease, so $L^0 > \omega + \delta$, and the value functions follow f_2^D. Therefore, we find the limit of f_2^D as $\bar{T}_A \to 0$. As $\bar{T}_A \to 0$, $k[L^0] - k_1[L^0]$ remains finite, but $\delta_1^D \to 0$, and $\delta k[L^0]$ approaches L^0. Therefore, the first two terms of f_2^D approach zero. The last term expands to

$$\bar{T}_A \left\lfloor \frac{L^0 - \omega}{v\bar{T}_A} \right\rfloor \left(v + C_H + C_N \frac{p}{1-p}\right).$$

As $\bar{T}_A \to 0$, this approaches

$$L^0 \left(1 + \frac{1}{v}\left(C_H + C_N \frac{p}{1-p}\right)\right). \tag{7.11}$$

Now, manipulation of Eq. (7.6) yields $J_{\sigma^\dagger}\left(\bar{T}_A\right) = f_2^D\left(L^0\right) + \bar{T}_A C_N \frac{p}{1-p}$. But as $\bar{T}_A \to 0$, the second term approaches zero. Thus, $J_{\sigma^\dagger}(\bar{T}_A)$ approaches Eq. (7.11). We have proved Eq. (7.7).

Large $\bar{T}_A$

There are several cases. First, consider $\delta < \omega$ and $\bar{T}_A \geq L^0/v$. The second condition implies that D keeps A in the first honeypot that he enters until all residual utility is exhausted, which produces utility $(v + C_H)L^0/v$. The first condition implies that $L^0/v > \bar{T}_\omega$, so $\bar{T}_A > \bar{T}_\omega$, which means that D ejects A from the first normal system that he enters, which produces 0 utility. The weighted sum of these utilities gives Eq. (7.8). Next, consider $\delta > \omega$ and $\bar{T}_A \geq L^0/\omega$. The first condition implies that $L^0/v < \bar{T}_\omega$, so it not guaranteed that $\bar{T}_A \geq \bar{T}_\omega$. But if $\bar{T}_A \geq \bar{T}_\omega$, D ejects A from the first normal system that he enters, and we have Eq. (7.8).

7.9 Notes

Reference [111] studies the belief of the attacker and suggests that the attacker should be ejected when he becomes suspicious that he may be in a honeypot. This is a useful complement to the present work. Other recent work has studied timing for more general interactions in cyber-physical systems [21, 136, 157] and network security in general [158]. On the contrary, we focus on timing in attacker engagement. This chapter also fits within the general category of optimal stopping problems. Optimal stopping problems with a finite horizon can be solved directly by dynamic programming, but our problem has an infinite horizon and is undiscounted.

Part III
Mitigation of Malicious Deception

Chapter 8
Strategic Trust

Part II studied defensive deception. It began with a taxonomy of defensive deception (Chap. 4). Then it described three models for the species defined within that taxonomy (Chaps. 5–7). Part III shifts the perspective and studies methods that a defender can use to mitigate malicious deception. In this sense, Part III studies a sort of dual problem to that of Part II.

In fact, models in Part III build upon the foundational models introduced in Part II, such as signaling games (Chaps. 3, 6, 9 and this chapter) and large population games (Chaps. 5 and 9). Part III also builds upon the theme of combining multiple games in order to study complex interactions (Chap. 5 and this chapter). One difference between the two parts is that while Part II aimed for a certain comprehensiveness through a literature review and taxonomy, Part III has no such aim. We limit ourselves to introducing models for the mitigation of malicious deception, while suggesting a taxonomy and literature review of this area for future work.

Within Part III, the current chapter studies *strategic trust*. Because of threats to the IoT, agents in the network must decide whether to trust other possibly-malicious agents, i.e., must assess their reliability and dependability. Unfortunately, the dynamic and plug-n-play nature of the IoT makes reputation-based trust systems insufficient. Hence, this chapter develops a framework for predictive or strategic trust in which agents make decisions in view of the incentives of the agents with whom they are communicating. These incentives are often related to the control of physical processes. Hence, this chapter quantifies utility based on the objective function of an optimal control problem.[1]

[1] For readers with more interest in optimal control than in game theory, the problem in this chapter can be stated differently: how should an observer-based optimal feedback control decide upon a subset of outputs to incorporate into the observer, given that some of the outputs may be compromised by an attacker?

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_8

We call the model in this chapter iSTRICT, an interdependent strategic trust mechanism for the cloud-enabled Internet of controlled things. iSTRICT is composed of three interdependent layers. In the cloud layer, iSTRICT uses a model called `FlipIt` games to conceptualize APTs. In the communication layer, it captures the interaction between devices and the cloud using signaling games. In the physical layer, we use optimal control to quantify the utilities in the higher level games. Best response dynamics link the three layers in an overall "game-of-games," for which the outcome is captured by a concept called Gestalt Nash equilibrium (GNE). The existence of a GNE is proved under a set of natural assumptions, and an adaptive algorithm is developed to iteratively compute the equilibrium. We then apply iSTRICT to trust management for autonomous vehicles that rely on measurements from remote sources. The chapter shows that strategic trust in the communication layer achieves a worst-case probability of compromise for any attack and defense costs in the cyber layer.

8.1 Strategic Trust for Mitigation of APTs

A key feature of the IoT is that it consists of physical devices which impact dynamic processes. The IoT offers opportunities for not only monitoring these systems, but also *controlling* them through networked sensors and actuators. This perspective on the IoT has given rise to the term "Internet of *controlled* things," or IoCT. Key features in the IoCT include those described for the IoT in Sect. 1.1, but with an emphasis on dynamics. The related concept of cyber-physical systems (CPS) refers to "smart networked systems with embedded sensors, processors, and actuators" [159]. Here, we take CPS and IoCT to have the same meaning. In these networks, "the joint behavior of the 'cyber' and physical elements of the system is critical—computing, control, sensing, and networking can be integrated into every component" [159].

The IoCT requires an interface between heterogeneous components. Local clouds (or *fogs* or *cloudlets*) offer promising solutions. In these networks, a cloud provides services for data aggregation, data storage, and computation. In addition, the cloud provides a market for the services of software developers and computational intelligence experts [7]. In these networks, sensors push environment data to the cloud, where it is aggregated and sent to devices (or "things"), which use the data for feedback control. These devices modify the environment, and the cycle continues. The control design of the IoCT is distributed, since each device can determine which cloud services to use for feedback control.

8.1.1 Advanced Persistent Threats in the Cloud-Enabled IoCT

Cyberattacks on the cloud are increasing as more businesses utilize cloud services [160]. To provide reliable support for IoCT applications, sensitive data provided by the cloud services needs to be well protected [161]. In this chapter, we focus on the attack model of APTs: "cyber attacks executed by sophisticated and well-resourced adversaries targeting specific information in high-profile companies and governments, usually in a long term campaign involving different steps" [142]. In the initial stage of an APT, an attacker penetrates the network through techniques such as social engineering, malicious hardware injection, theft of cryptographic keys, or zero-day exploits [162]. For example, the *Naikon APT*, which targeted governments around the South China Sea in 2010–2015, used a bait document that appeared to be a Microsoft Word file but which was actually a malicious executable that installed spyware [163].

The cloud is particularly vulnerable to initial penetration through attacks on the application layer, because many applications are required for developers and clients to interface with the cloud. Cross-site scripting (XSS) and SQL injection are two types of application-layer attacks. In SQL injection, attackers insert malicious SQL code into fields which do not properly process string literal escape characters. The malicious code targets the server, where it could be used to modify data or bypass authentication systems. By contrast, XSS targets the execution of code in the browser on the client side. Both of these attacks give attackers an initial entry point into a system, from which they can begin to gain more complete, insider control. This control of the cloud can be used to transmit malicious signals to a CPS and cause physical damage.

8.1.2 Strategic Trust

Given the threat of insider attacks on the cloud, each IoCT device must decide which signals to trust from cloud services. Trust refers to positive beliefs about the perceived reliability of, dependability of, and confidence in another entity [164]. These entities may be agents in an IoCT with misaligned incentives. Many specific processes in the IoCT require trust, such as data collection, aggregation and processing, privacy protection, and user-device trust in human-in-the-loop interactions [165].

While many factors influence trust, including subjective beliefs, we focus on the objective properties of trust. These include (1) reputation, (2) promises, and (3) interaction context. Many trust management systems are based on tracking reputation over multiple interactions. Unfortunately, agents in the IoCT may interact only once, making reputation difficult to accrue [46]. This property of IoCT also limits the effectiveness of promises such as contracts or policies. Promises may not be enforceable for entities that interact only once. Therefore, we focus on strategic trust that is

predictive rather than reactive. We use game-theoretic utility functions to capture the motivations for entities to be trustworthy. These utility functions change based on the particular context of the interaction. In this sense, our model of strategic trust is *incentive-compatible*, i.e., consistent with each agent acting in its own self-interest.

8.1.3 Game-Theoretic iSTRICT Model

We propose a framework called iSTRICT, which is composed of three interacting layers: a cloud layer, a communication layer, and a physical layer. In the first layer, the cloud services are threatened by attackers capable of APTs and defended by network administrators (or "defenders"). The interaction at each cloud-service is modeled using the `FlipIt` game recently proposed by Bowers et al. [162] and van Dijk et al. [158]. iSTRICT uses one `FlipIt` game per cloud-service. In the communication layer, the cloud services—which may be controlled by the attacker or defender according to the outcome of the `FlipIt` game—transmit information to a device which decides whether to trust the cloud services. This interaction is captured using a signaling game. At the physical layer, the utility parameters for the signaling game are determined using optimal control. The cloud, communication, and physical layers are interdependent. This motivates an overall equilibrium concept called GNE. GNE requires each game to be solved optimally given the results of the other games. Because this is a similar idea to *best response* in Nash equilibrium, we call the multi-game framework a *game-of-games*.

The rest of the chapter proceeds as follows. In Sect. 8.2, we give a broad outline of the iSTRICT model. Section 8.3 presents the details of the `FlipIt` game, signaling game, physical layer control system, and equilibrium concept. Then, in Sect. 8.4, we study the equilibrium analytically using an adaptive algorithm. Finally, we apply the framework to the control of autonomous vehicles in Sect. 8.5.

8.2 iSTRICT Overview

Consider a cyber-physical attack in which an adversary penetrates a cloud service in order to transmit malicious signals to a physical device and cause improper operation. This type of cross-layer attack is increasingly relevant in IoCT settings. Perhaps the most famous cross-layer attack is the Stuxnet attack that damaged Iran's nuclear program. But even more recently, an attacker allegedly penetrated the SCADA system that controls the Bowman Dam, located less than 20 miles north of Manhattan. The attacker gained control of a sluice gate which manages water level[2] [166]. Cyber-

[2]The sluice gate happened to be disconnected for manual repair at the time, however, so the attacker could not actually change water levels.

physical systems ranging from the smart grid to public transportation need to be protected from similar attacks.

The iSTRICT framework offers a defense-in-depth approach to IoCT security. In this section, we introduce each of the three layers of iSTRICT very briefly, in order to focus on the interaction between the layers. We describe an equilibrium concept for the simultaneous steady-state of all three layers. Later, Sect. 8.3 describes each layer in detail. Table 8.1 lists the notation for the chapter.

8.2.1 Cloud Layer

Consider a cloud-enabled IoCT composed of sensors that push data to a cloud, which aggregates the data and sends it to devices. For example, in a cloud-enabled smart home, sensors could include lighting sensors, temperature sensors, and blood pressure or heart rate sensors that may be placed on the skin or embedded within the body. Data from these sensors is processed by a set of cloud services $\mathbb{S} = \{1, \ldots, N\}$, which make data available for control.

For each cloud service $i \in \mathbb{S}$, let A^i denote an attacker who attempts to penetrate the service using zero-day exploits, social engineering, or other techniques described in Sect. 8.1. Similarly, let D^i denote a defender or network administrator attempting to maintain the security of the cloud service. A^i and D^i attempt to claim or reclaim the control of each cloud service at periodic intervals. We model the interactions at all of the services using `FlipIt` games, one for each of the N services.

In the `FlipIt` game [158, 162], an attacker and a defender gain utility proportional to the amount of time that they control a resource (here a cloud service), and pay attack costs proportional to the number of times that they attempt to claim or reclaim the resource. We consider a version of the game in which the attacker and defender are restricted to attacking at fixed frequencies. The equilibrium of the game is a Nash equilibrium.

Let $v_A^i \in \mathbb{R}$ and $v_D^i \in \mathbb{R}$ denote the values of each cloud service $i \in \mathbb{S}$ to A^i and D^i, respectively. These quantities represent the inputs of the `FlipIt` game. The outputs of the `FlipIt` game are the proportions of time for which A^i and D^i control the cloud service. Denote these proportions by $p_A^i \in [0, 1]$ and $p_D^i = 1 - p_A^i$, respectively. To summarize each of the `FlipIt` games, define a set of mappings $T^{F_i} : \mathbb{R} \times \mathbb{R} \to [0, 1]$, $i \in \mathbb{S}$, such that

$$p_A^{i*} = T^{F_i}\left(v_A^i, v_D^i\right) \tag{8.1}$$

maps the values of cloud service i for A^i and D^i to the proportion of time p_A^{i*} for which the service will be compromised in equilibrium. We will study this mapping further in Sect. 8.3.1.

Table 8.1 Nomenclature for this chapter

Notation	Meaning
$\mathbb{S} = \{1, 2, \ldots, N\}$	Set of cloud services (CSs)
A^i, D^i, $i \in \mathbb{S}$, R	Attackers and defenders, cloud-enabled device
$v_A = [v_A^i]_{i\in\mathbb{S}}$, $v_D = [v_D^i]_{i\in\mathbb{S}}$	Values of CSs for A and D
$p_A = [p_A^i]_{i\in\mathbb{S}}$, $p_D = [p_D^i]_{i\in\mathbb{S}}$	Probabilities that A and D control CSs
$p_A^{i*} = T^{F_i}(v_A^i, v_D^i)$	`FlipIt` mapping for CS i
$(v_A^*, v_D^*) \in T^S(p_A)$	Signaling game mapping
f_A^i, f_D^i	Attack and recapture frequencies for CS i
$U_A^{F_i}(f_A^i, f_D^i)$, $U_D^{F_i}(f_A^i, f_D^i)$	Utilities of A^i and D^i in `FlipIt` game i
$\theta = [\theta^i]_{i\in\mathbb{S}}$	Types of CSs
$\theta^i \in \Theta = \{\theta_A, \theta_D\}$	Space of types: either attacker- or defender-controlled
$m = [m^i]_{i\in\mathbb{S}}$	Messages from CSs
$m^i \in \mathbb{M} = \{m_L, m_H\}$	Space of messages: either low or high-risk
$a = [a^i]_{i\in\mathbb{S}}$	Actions for CSs
$a^i \in \mathbb{A} = \{a_T, a_N\}$	Space of actions: either trust or not trust
$u_A^{S_i}(m, a)$, $u_D^{S_i}(m, a)$	Signaling game utility functions for A^i and D^i
$u_R^S(\theta, m, a)$	Signaling game utility function for R
$\sigma_A^i(m) \in \Sigma_A$, $\sigma_D^i(m) \in \Sigma_D$	Signaling game mixed strategies for A^i and D^i
$\sigma_R(a \mid m) \in \Sigma_R^N$	Signaling game mixed strategy for R
$\mu(\theta \mid m) = [\mu^i(\theta \mid m)]_{i\in\mathbb{S}}$	Beliefs of R CSs
$U_A^{S_i}(\sigma_R; \sigma_A^i, \sigma_A^{-i}; \sigma_D^{-i})$	Signaling game mixed-strategy utility for A^i
$U_D^{S_i}(\sigma_R; \sigma_A^{-i}; \sigma_D^i, \sigma_D^{-i})$	Signaling game mixed-strategy utility for D^i
$x[k]$, $\hat{x}[k]$, $u[k]$, $w[k]$	State, estimated state, control, and process noise
$\Delta_A^i[k]$, $\Delta_D^i[k]$, $\Xi_\theta[k]$	Attacker and defender bias terms, cloud type matrix
$y[k]$, $\tilde{y}[k]$, $v[k]$	Measurements without and with biases, sensor noise
ξ, ζ	Covariance matrices of process and sensor noise
$\nu[k]$, ϵ	Innovation and vector of innovation thresholds
$D_{\sigma_R}(\nu[k])$	Innovation gate
$v_{AD}^i = v_A^i / v_D^i$, $i \in \mathbb{S}$	Ratios of values of CSs for A^i and D^i
$\mathbb{V}^i$, $\mathbb{PR}^i$	Spaces of possible v_{AD}^i and p_A^i in a GNE
$p_A^{i*} = \tilde{T}^{F_i}(v_{AD}^i)$	Redefined `FlipIt` mapping for CS i
$v_{AD}^* \in \tilde{T}^S(p_A)$	Redefined signaling game mapping
$v_{AD}^* \in \tilde{T}^{S\circ F}(v_{AD})$	Composition of $\tilde{T}^{F_i}$, $i \in \mathbb{S}$ and $\tilde{T}^S$
$v_{AD}^\dagger \in \tilde{T}^{S\circ F}(v_{AD}^\dagger)$	Fixed-point requirement for a GNE

8.2.2 Communication Layer

In the *communication layer*, the cloud services $i \in \mathbb{S}$, which each may be controlled by A^i or D^i, send data to a device R, which decides whether to trust the signals. This interaction is modeled by a signaling game. The signaling game *sender* is the cloud service. The two *types* of the sender are attacker or defender. The signaling game *receiver* is the device R. While we used N `FlipIt` games to describe the cloud layer, we use only one signaling game to describe the communication layer, because R must decide which services to trust all at once.

The prior probabilities in the communication layer are the equilibrium proportions p_A^i and $p_D^i = 1 - p_A^i$, $i \in \mathbb{S}$ from the equilibrium of the cloud layer. Denote the vectors of the prior probabilities for each sensor by $p_A = \left[p_A^i\right]_{i\in\mathbb{S}}$, $p_D = \left[p_D^i\right]_{i\in\mathbb{S}}$. These prior probabilities are the inputs of the signaling game.

The outputs of the signaling game are the equilibrium utilities received by the senders. Denote these utilities by v_A^i and v_D^i, $i \in \mathbb{S}$. Importantly, these are the same quantities that describe the incentives of A and D to control each cloud service in the `FlipIt` game, because the party which controls each service is awarded the opportunity to be the sender in the signaling game. Define vectors to represent each of these utilities by $v_A = \left[v_A^i\right]_{i\in\mathbb{S}}$, $v_D = \left[v_D^i\right]_{i\in\mathbb{S}}$.

Finally, let $T^S : [0, 1]^N \to \mathcal{P}(\mathbb{R}^{2N})$ be a mapping that summarizes the signaling game, where $\mathcal{P}(\mathbb{X})$ is the power set of $\mathbb{X}$. According to this mapping, the set of vectors of signaling game equilibrium utility ratios v_A^* and v_D^* that result from the vector of prior probabilities p_A is given by

$$\left(v_A^*, v_D^*\right) = T^S\left(p_A\right). \tag{8.2}$$

This mapping summarizes the signaling game. We study the mapping in detail in Sect. 8.3.2.

8.2.3 Physical Layer

Many IoCT devices such as pacemakers, cleaning robots, appliances, and electric vehicles are dynamic systems that operate using feedback mechanisms. The physical-layer control of these devices requires remote sensing of the environment and the data stored or processed in the cloud. The security at the cloud and the communication layers of the system are intertwined with the performance of the controlled devices at the physical layer. Therefore, the trustworthiness of the data has a direct impact on the control performance of the devices. This control performance determines the utility of the device R as well as the utility of each of the attackers A^i and defenders D^i. The control performance is quantified using a cost criterion for observer-based optimal feedback control. The observer uses data from the cloud services that R

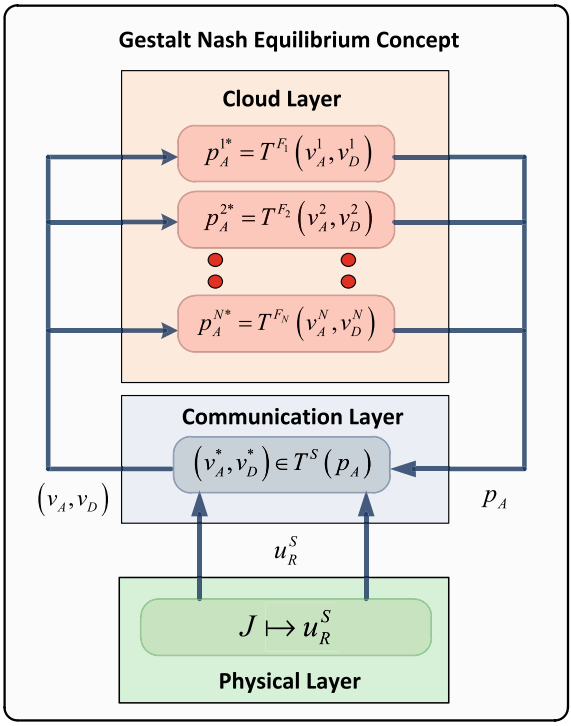


Fig. 8.1 In iSTRICT, `FlipIt` games model attacks on the set of cloud services. T^{F_i}, $i \in \mathbb{S}$, map the value of each service to the proportion of time that it will be compromised in equilibrium. The communication layer is modeled by a signaling game. T^S maps probabilities of compromise to the value of each cloud service. The cloud layer and communication layer are interdependent. The physical layer performance quantifies the utilities for the signaling game

elects to trust, and ignores the cloud services that R decides not to trust. We study the physical layer control in Sect. 8.3.3.

8.2.4 *Coupling of the Cloud and Communication Layers*

Clearly, the cloud and communication layers are coupled through Eqs. (8.1) and (8.2). The cloud layer security serves as an input to the communication layer. The resulting utilities of the signaling game at the communication layer further become an input to the FlipIt game at the cloud layer. In addition, the physical layer performance quantifies the utilities for the signaling games. Figure 8.1 depicts this concept. In order to predict the behavior of the whole cloud-enabled IoCT, iSTRICT considers an equilibrium concept which we call GNE. Informally, a triple $\left(p_A^\dagger, v_A^\dagger, v_D^\dagger\right)$ is a GNE if it simultaneously satisfies Eqs. (8.1) and (8.2).

GNE is useful for three reasons. First, cloud-enabled IoCT networks are dynamic. The modular structure of GNE requires the `FlipIt` games and the signaling game to be at equilibrium given the parameters that they receive from the other type of game. In extensive-form games, this requirement is called *subgame perfection*: the equilibrium solution for the overall game must also be an equilibrium solution for

each subgame that it contains [90]. In GNE, however, perfection applies in both directions, because there is no clear chronological order or directional flow of information between the two games. Actions in each subgame must be chosen by prior commitment relative to the results of the other subgame.

Second, GNE draws upon established results from `FlipIt` games and signaling games instead of attempting to analyze one large game. IoCT networks promise plug-and-play capabilities, in which devices and users are easily able to enter and leave the network. This also motivates the plug-and-play availability of solution concepts. The solution to one subgame should not need to be totally recomputed if an actor enters or leaves another subgame. GNE follows this approach.

Finally, GNE serves as an example of a solution approach which could be called *game-of-games*. The equilibrium solutions to the `FlipIt` games and signaling game must be rational "best responses" to the solution of the other type of game.

8.3 Detailed iSTRICT Model

In this section, we define more precisely the three layers of the iSTRICT framework.

8.3.1 Cloud Layer: `FlipIt` Game

We use a `FlipIt` game to model the interactions between the attacker and the defender over each cloud service.

`FlipIt` **Actions**

For each service, A^i and D^i choose f_A^i and f_D^i, the frequencies with which they claim or reclaim control of the service. These frequencies are chosen by prior commitment. Neither player knows the other player's action when she makes her choice. Figure 8.2 depicts the `FlipIt` game. The green boxes above the horizontal axis represent control of the service by D^i and the red boxes below the axis represent control of the service by A^i.

From f_A^i and f_D^i, it is easy to compute the expected proportions of the time that A and D control service i [158, 162]. Let $\mathbb{R}_+$ denote the set of nonnegative real numbers. Define the function $\rho : \mathbb{R}_+ \times \mathbb{R}_+ \to [0, 1]$, such that $p_A^i = \rho\left(f_A^i, f_D^i\right)$ gives the proportion of the time that A^i will control the cloud service if he attacks with frequency f_A^i and D^i renews control of the service (through changing cryptographic keys or passwords, or through installing new hardware) with frequency f_D^i. We have

$$\rho\left(f_A^i, f_D^i\right) = \begin{cases} 0, & \text{if } f_A^i = 0, \\ \frac{f_A^i}{2f_D^i}, & \text{if } f_D^i \geq f_A^i > 0, \\ 1 - \frac{f_D^i}{2f_A^i}, & \text{if } f_A^i > f_D^i \geq 0. \end{cases} \tag{8.3}$$

Notice that when $f_A^i > f_D^i \geq 0$, i.e., the attacking frequency of A^i is greater than the renewal frequency of D^i, the proportion of time that service i is insecure is $\rho\left(f_A^i, f_D^i\right) > \frac{1}{2}$, and when $f_D^i \geq f_A^i > 0$, we obtain $\rho\left(f_A^i, f_D^i\right) \leq \frac{1}{2}$.

`FlipIt` **Utility Functions**

Recall that v_A^i and v_D^i denote the value of controlling service $i \in \mathbb{S}$ for A^i and D^i, respectively. These quantities define the heights of the red and green boxes in Fig. 8.2. Denote the costs of renewing control of the cloud service for the two players by α_A^i and α_D^i. Finally, let $U_A^{F_i} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ and $U_D^{F_i} : \mathbb{R}_+ \times \mathbb{R}_+ \to \mathbb{R}$ be expected utility functions for each `FlipIt` game. The utilities of each player are given in Eqs. (8.4) and (8.5) by the values v_D^i and v_A^i of controlling the service multiplied by the proportions p_D^i and p_A^i with which the service is controlled, minus the costs α_D^i and α_A^i of attempting to renew control of the service.

$$U_D^{F_i}\left(f_A^i, f_D^i\right) = v_D^i\left(1 - \rho\left(f_A^i, f_D^i\right)\right) - \alpha_D^i f_D^i. \tag{8.4}$$

$$U_A^{F_i}\left(f_A^i, f_D^i\right) = v_A^i \rho\left(f_A^i, f_D^i\right) - \alpha_A^i f_A^i. \tag{8.5}$$

Therefore, based on the attacker's action f_A^i, the defender determines f_D^i strategically to maximize the proportional time of controlling the cloud service i, $1 - \rho(f_A^i, f_D^i)$, and minimize the cost of choosing f_D^i.

Note that in the game, the attacker knows v_A^i and α_A^i, and the defender knows v_D^i and α_D^i. Furthermore, $\rho(f_A^i, f_D^i)$ is public information, and hence both players know the frequencies of control of the cloud through (8.3). Therefore, the communication between two players at the cloud layer is not necessary when determining their strategies.

`FlipIt` **Equilibrium Concept**

The equilibrium concept for the `FlipIt` game is Nash equilibrium, since it is a complete information game in which strategies are chosen by prior commitment.

Definition 8.1 (*Nash Equilibrium*) A Nash equilibrium of the `FlipIt` game played for control of service $i \in \{1, \ldots, N\}$ is a strategy profile $\left(f_A^{i*}, f_D^{i*}\right)$ such that

$$f_D^{i*} \in \underset{f_D^i \in \mathbb{R}_+}{\arg\max}\, U_D^{F_i}\left(f_A^{i*}, f_D^{i*}\right), \tag{8.6}$$

$$f_A^{i*} \in \underset{f_A^i \in \mathbb{R}_+}{\arg\max}\, U_A^{F_i}\left(f_A^{i*}, f_D^{i*}\right), \tag{8.7}$$

where $U_D^{F_i}$ and $U_A^{F_i}$ are computed by Eqs. (8.4) and (8.5).

From the equilibrium frequencies f_D^{i*} and f_A^{i*}, let the equilibrium proportion of time that A^i controls cloud service i be given by p_A^{i*} according to Eq. (8.3). The Nash equilibrium solution can then be used to determine the mapping in Eq. (8.1) from the

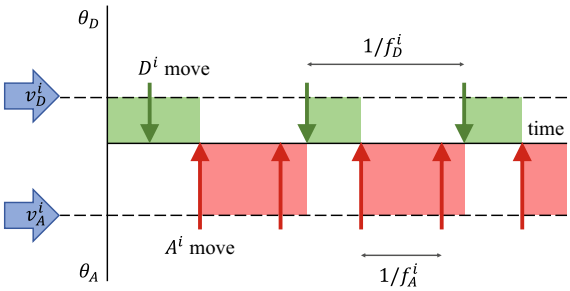


Fig. 8.2 In each `FlipIt` game, A^i and D^i periodically claim control of cloud service i. The values of the service for each player are given by v_A^i and v_D^i, which depend on the equilibrium of the signaling game

cloud service values v_A^i and v_D^i to the equilibrium attacker control proportion p_A^{i*}, where $T^{F_i} : \mathbb{R} \times \mathbb{R} \to [0, 1]$. The T^{F_i} mappings, $i \in \mathbb{S}$, constitute the top layer of Fig. 8.1.

8.3.2 *Communication Layer: Signaling Game*

Because the cloud services are vulnerable, devices which depend on data from the services should rationally decide whether to trust them. This is captured using a signaling game. In this model, the device R updates a belief about the state of each cloud service and decides whether to trust it. Figure 8.3 depicts the actions that correspond to one service of the signaling game. Compared to trust value-based trust management systems, in which reputation attacks can significantly influence trust decisions [167, 168], in iSTRICT, R's decision is based on the strategies of each A^i and D^i at the cloud layer as well as the physical layer performance, and hence it does not depend on the feedback of cloud services from users, which could be malicious. We next present the detailed model of the signaling game.

Signaling Game Types

The *types* of each cloud service $i \in \mathbb{S}$ are $\theta^i \in \Theta = \{\theta_A, \theta_D\}$, where $\theta^i = \theta_A$ indicates that the service is compromised by A^i, and $\theta = \theta_D$ indicates that the service is controlled by D^i. Denote the vector of all the service types by $\theta = \left[\theta^i\right]_{i\in\mathbb{S}} \triangleq \left[\theta^1\ \theta^2\ \ldots\ \theta^m\right]^T$.

Signaling Game Messages

Denote the risk level of the data from each service i by $m^i \in M = \{m_L, m_H\}$, where m_L and m_H indicate low-risk and high-risk messages, respectively. (We define this risk level in Sect. 8.3.3.) Further, define the vector of all of the risk levels by $m = \left[m^i\right]_{i\in\mathbb{S}}$.

Next, define mixed strategies for A^i and D^i. Let $\sigma_A^i : M \to [0, 1]$ and $\sigma_D^i : M \to [0, 1]$ be functions such that $\sigma_A^i\left(m_A^i\right) \in \Sigma_A$ and $\sigma_D^i\left(m_D^i\right) \in \Sigma_D$ give the proportions with which A^i and D^i send messages with risk levels m_A^i and m_D^i, respectively, from each cloud service i that they control. Note that R only observes m_A^i or m_D^i, depending on who controls the service i. Let

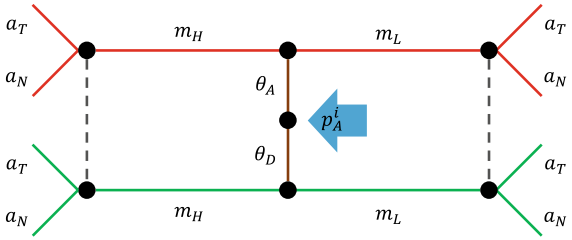


Fig. 8.3 The vector of types $\theta \in \Theta^N$ defines whether each cloud service $i \in \mathbb{S}$ is controlled by an attacker or defender. Each prior probability p_A^i comes from the corresponding `FlipIt` game. The player who controls each service chooses m^i. R observes all m^i and chooses a^i, $i \in \mathbb{S}$ simultaneously. Here, we show one service, although all of the services are coupled

$$m^i = \begin{cases} m_A^i, & \text{if} \quad \theta^i = \theta_A \\ m_D^i, & \text{if} \quad \theta^i = \theta_D \end{cases},$$

denote risk level of the message that R actually observes. Finally, define the vector of observed risk levels by $m = \left[m^i\right]_{i \in \mathbb{S}}$.

Signaling Game Beliefs and Actions

Based on the risk levels m that R observes, it updates its vector of prior beliefs p_A. Define $\mu^i : \Theta \to [0, 1]$, such that $\mu^i\left(\theta \mid m^i\right)$ gives the belief of R that service $i \in \mathbb{S}$ is of type θ given that R observes risk level m^i. Also write the vector of beliefs as $\mu\left(\theta \mid m\right) = \left[\mu^i\left(\theta^i \mid m^i\right)\right]_{i \in \mathbb{S}}$. As a direction for future work, we note that evidence-based signaling game approaches could be used to update belief in a manner robust to reputation attacks [135, 140, 155].

Based on these beliefs, R chooses which cloud services to trust. For each service i, R chooses $a^i \in A = \{a_T, a_N\}$ where a_T denotes trusting the service (i.e., using it for observer-based optimal feedback control) and a_N denotes not trusting the service. Assume that R, aware of the system dynamics, chooses actions for each service simultaneously, i.e., $a = \left[a^i\right]_{i \in \mathbb{S}}$.

Next, define $\sigma_R : A^N \to [0, 1]$ such that $\sigma_R\left(a \mid m\right) \in \Sigma_R^N$ gives the mixed-strategy probability with which R plays the vector of actions a given the vector of risk levels m.

Signaling Game Utility Functions

Let R's utility function be denoted by $u_R^S : \Theta^N \times M^N \times A^N \to \mathbb{R}$, such that $u_R^S\left(\theta, m, a\right)$ gives the utility that R receives when θ is the vector of cloud service types, m is the vector of risk levels, and R chooses the vector of actions a.

For $i \in \mathbb{S}$, define the functions $u_A^{S_i} : M^N \times A^N \to \mathbb{R}$ and $u_D^{S_i} : M^N \times A^N \to \mathbb{R}$, such that $u_A^{S_i}\left(m, a\right)$ and $u_D^{S_i}\left(m, a\right)$ give the utility that A^i and D^i receive for service i when the risk levels are given by the vector m, and R plays the vector of actions a.

Next, consider expected utilities based on the strategies of each player. Let $U_R^S : \Sigma_R^N \to \mathbb{R}$ denote the expected utility function for R, such that $U_R^S\left(\sigma_R \mid m, \mu\left(\bullet \mid m\right)\right)$

gives R's expected utility when he plays mixed strategy σ_R given that he observes risk levels m and has belief μ. We have

$$U_R^S(\sigma_R \mid m, \mu) = \sum_{\theta \in \Theta^m} \sum_{a \in A^m} u_R^S(\theta, m, a)\, \mu(\theta \mid m)\, \sigma_R(a \mid m). \tag{8.8}$$

In order to compute the expected utility functions for A^i and D^i, define $\sigma_A^{-i} = \left\{\sigma_A^j \mid j \in \mathbb{S}\backslash\{i\}\right\}$ and $\sigma_D^{-i} = \left\{\sigma_D^j \mid j \in \mathbb{S}\backslash\{i\}\right\}$, the sets of the strategies of all of the senders except the sender on cloud service i. Then define $U_A^{S_i} : \Sigma_R^N \times \Sigma_A^N \times \Sigma_D^{N-1} \to \mathbb{R}$ such that $U_A^{S_i}\left(\sigma_R; \sigma_A^i, \sigma_A^{-i}; \sigma_D^{-i}\right)$ gives the expected utility to A^i when he plays mixed strategy σ_A^i, and the attackers and defenders on the other services play σ_A^{-i} and σ_D^{-i}. Define the expected utility to D^i by $U_D^{S_i}\left(\sigma_R; \sigma_A^{-i}; \sigma_D^i, \sigma_D^{-i}\right)$ in a similar manner.

Let $X^i \in \{A, D\}$ denote the player that controls service i and $X \in \{A, D\}^N$ denote the set of players that control each service. Then the expected utilities are computed by

$$\begin{aligned} U_A^{S_i}\left(\sigma_R; \sigma_A^i, \sigma_A^{-i}; \sigma_D^{-i}\right) = & \sum_{m \in M^N} \sum_{a \in A^N} \\ & \sum_{X^1 \in \{A,D\}} \cdots \sum_{X^{i-1} \in \{A,D\}} \sum_{X^{i+1} \in \{A,D\}} \cdots \sum_{X^N \in \{A,D\}} \\ & u_A^{S_i}(m, a)\, \sigma_R(a \mid m)\, \sigma_A^i\left(m^i\right) \prod_{j \in \mathbb{S}\backslash\{i\}} \sigma_{X^j}^j\left(m^j\right) p_{X^j}^j, \end{aligned} \tag{8.9}$$

$$\begin{aligned} U_D^{S_i}\left(\sigma_R; \sigma_A^{-i}; \sigma_D^i, \sigma_D^{-i}\right) = & \sum_{m \in M^N} \sum_{a \in A^N} \\ & \sum_{X^1 \in \{A,D\}} \cdots \sum_{X^{i-1} \in \{A,D\}} \sum_{X^{i+1} \in \{A,D\}} \cdots \sum_{X^N \in \{A,D\}} \\ & u_D^{S_i}(m, a)\, \sigma_R(a \mid m)\, \sigma_D^i\left(m^i\right) \prod_{j \in \mathbb{S}\backslash\{i\}} \sigma_{X^j}^j\left(m^j\right) p_{X^j}^j. \end{aligned} \tag{8.10}$$

Perfect Bayesian Nash Equilibrium Conditions

Finally, we can state the requirements for a PBNE for the signaling game [90].

Definition 8.2 (*PBNE*) For the device, let $U_R^S(\sigma_R \mid m, \mu)$ be formulated according to Eq. (8.8). For each service $i \in \mathbb{S}$, let $U_A^{S_i}\left(\sigma_R; \sigma_A^i, \sigma_A^{-i}; \sigma_D^{-i}\right)$ be given by Eq. (8.9) and $U_D^{S_i}\left(\sigma_R; \sigma_A^{-i}; \sigma_D^i, \sigma_D^{-i}\right)$ be given by Eq. (8.10). Finally, let vector p_A give the prior probabilities of each service being compromised. Then, a PBNE of the signaling game is a strategy profile $\left(\sigma_R^*; \sigma_A^{1*}, \ldots \sigma_A^{N*}; \sigma_D^{1*}, \ldots \sigma_D^{N*}\right)$ and a vector of beliefs $\mu(\theta \mid m)$ such that the following hold:

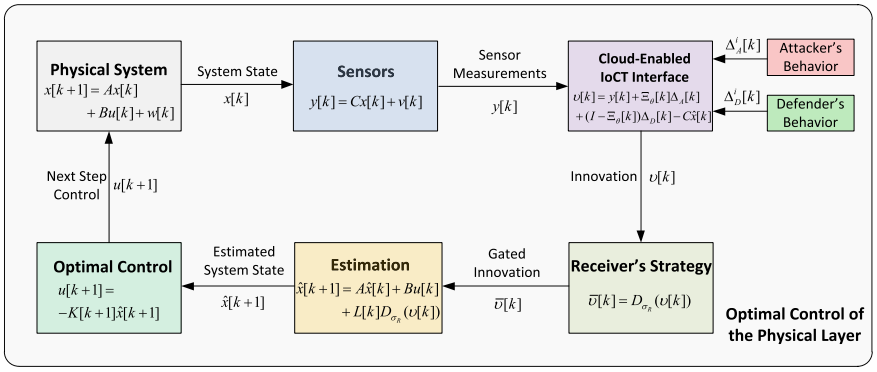


Fig. 8.4 A block diagram shows the various components of the control system in the iSTRICT. The physical system refers to IoCT device whose states are collected by sensors. Each A^i and D^i in the cloud layer can add bias terms to the measured sensor data before sending it to the receiver. R decides whether to trust or not trust each of the cloud services and then designs an optimal control for the physical system. Since the optimal control is designed over a finite-horizon cost criterion, the loop terminates after T time steps

$$\forall i \in \mathbb{S},\ \sigma_A^{i*}(\bullet) \in \underset{\sigma_A^i \in \Sigma_A}{\arg\max}\, U_A^{S_i}\left(\sigma_R^*; \sigma_A^i, \sigma_A^{-i*}; \sigma_D^{-i*}\right), \tag{8.11}$$

$$\forall i \in \mathbb{S},\ \sigma_D^{i*}(\bullet) \in \underset{\sigma_D^i \in \Sigma_D}{\arg\max}\, U_D^{S_i}\left(\sigma_R^*; \sigma_A^{-i*}; \sigma_D^i, \sigma_D^{-i*}\right), \tag{8.12}$$

$$\forall m \in M,\ \sigma_R^* \in \underset{\sigma_R \in \Sigma_R^m}{\arg\max}\, U_R^S\left(\sigma_R \mid m, \mu(\bullet \mid m)\right), \tag{8.13}$$

and $\forall i \in \mathbb{S}$,

$$\mu^i\left(\theta_A \mid m^i\right) = \frac{\sigma_A^{i*}\left(m^i\right) p_A^i}{\sigma_A^{i*}\left(m^i\right) p_A^i + \sigma_D^{i*}\left(m^i\right)\left(1 - p_A^i\right)}, \tag{8.14}$$

if $\sigma_A^{i*}\left(m^i\right) p_A^i + \sigma_D^{i*}\left(m^i\right) p_D^i \neq 0$, and $\mu^i\left(\theta_A \mid m^i\right) \in [0,1]$, if $\sigma_A^{i*}\left(m^i\right) p_A^i + \sigma_D^{i*}\left(m^i\right) p_D^i = 0$. Additionally, $\mu^i\left(\theta_D \mid m^i\right) = 1 - \mu^i\left(\theta_A \mid m^i\right)$ in both cases.

Note that we have denoted the equilibrium utilities for A^i and D^i, $i \in \mathbb{S}$ by

$$v_A^i = U_A^{S_i}\left(\sigma_R^*; \sigma_A^{i*}, \sigma_A^{-i*}; \sigma_D^{-i*}\right), \tag{8.15}$$

$$v_D^i = U_D^{S_i}\left(\sigma_R^*; \sigma_A^{-i*}; \sigma_D^{i*}, \sigma_D^{-i*}\right), \tag{8.16}$$

and the vectors of those values by $v_A = \left[v_A^i\right]_{i\in\mathbb{S}}$, $v_D = \left[v_D^i\right]_{i\in\mathbb{S}}$. We now have the complete description of the signaling game mapping Eq. (8.2), where $T^S : [0,1]^N \to \mathcal{P}(\mathbb{R}^{2N})$. This mapping constitutes the middle layer of Fig. 8.1.

8.3.3 Physical Layer: Optimal Control

The utility function $u_R^S(\theta, m, a)$ is determined by the performance of the device controller as shown in Fig. 8.1. A block illustration of the control system is shown in Fig. 8.4. Note that the physical system in the diagram refers to the IoCT devices.

Device Dynamics

Each device in the IoCT is governed by dynamics. We can capture the dynamics of the things by the linear system model

$$x[k+1] = Ax[k] + Bu[k] + w[k], \tag{8.17}$$

where $A \in \mathbb{R}^{n\times n}$, $B \in \mathbb{R}^{n\times q}$, $x[k] \in \mathbb{R}^n$ is the system state, $u[k] \in \mathbb{R}^q$ is the control input, $w[k]$ denotes the system white noise, and $x[0] = x_0 \in \mathbb{R}^n$ is given. Let $y[k] \in \mathbb{R}^N$ represent data from cloud services which suffers from white, additive Gaussian sensor noise given by the vector $v[k]$. We have $y[k] = Cx[k] + v[k]$, where $C \in \mathbb{R}^{N\times n}$ is the output matrix. Let the system and sensor noise processes have known covariance matrices $\mathbb{E}\left\{w[k]w'[k]\right\} = \xi$, $\mathbb{E}\left\{v[k]v'[k]\right\} = \zeta$, where ξ and ζ are symmetric, positive, semi-definite matrices, and $w'[k]$ and $v'[k]$ denote the transposes of the noise vectors.

In addition, for each cloud service $i \in \mathbb{S}$, the attacker A^i and defender D^i in the signaling game choose whether to add bias terms to the measurement $y^i[k]$. Let $\Delta_A^i[k]$, $\Delta_D^i[k] \in \mathbb{R}$ denote these bias terms. The actual noise levels that R observes depends on who controls the service in the `FlipIt` game. Recall that the vector of types of each service is given by $\theta = \left[\theta^i\right]_{i\in\mathbb{S}}$. Let $\mathbf{1}_{\{\bullet\}}$ represent the indicator function, which takes the value of 1 if its argument is true and 0 otherwise. Then, define the matrix

$$\Xi_\theta = \text{diag}\left\{\left[\mathbf{1}_{\{\theta^1=\theta_A\}} \cdots \mathbf{1}_{\{\theta^N=\theta_A\}}\right]\right\}.$$

Including the bias term, the measurements are given by

$$\tilde{y}[k] = Cx[k] + v[k] + \Xi_\theta[k]\Delta_A[k] + (I - \Xi_\theta[k])\,\Delta_D[k], \tag{8.18}$$

where I is the N-dimensional identity matrix.

Observer-Based Optimal Feedback Control

Let F, Q, and R be positive-definite matrices of dimensions $n \times n$, $n \times n$, and $q \times q$, respectively. The device chooses the control u that minimizes the operational cost given by

$$J = \mathbb{E}\left\{x'[T]Fx[T] + \sum_{k=0}^{T-1} x'[k]Qx[k] + u'[k]Ru[k]\right\}, \tag{8.19}$$

subject to the dynamics of Eq. (8.17).

To attempt to minimize Eq. (8.19), the device uses *observer-based optimal feedback control*. Define $P[k]$ by the forward Riccati difference equation

$$P[k+1] = A\left(P[k] - P[k]C'\left(CP[k]C' + \xi\right)^{-1} CP[k]\right)A' + \zeta,$$

with $P[0] = \mathbb{E}\{(x[0] - \hat{x}[0])(x[0] - \hat{x}[0])'\}$, and let $L[k] = P[k]C'(CP[k]C' + \xi)^{-1}$. Then the observer is a Kalman filter given by [169]

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] + L[k]\left(\tilde{y}[k] - C\hat{x}[k]\right).$$

Innovation

In this context, the term $\tilde{y}[k] - C\hat{x}[k]$ is the *innovation*. Label the innovation by $\nu[k] = \tilde{y}[k] - C\hat{x}[k]$. This term is used to update the estimate $\hat{x}[k]$ of the state. We consider the components of the innovation as the signaling-game *messages* that the device decides whether to trust. Let us label each component of the innovation as low-risk or high-risk. For each $i \in \mathbb{S}$, we classify the innovation as

$$m^i = \begin{cases} m_L, & \text{if } \left|\nu^i[k]\right| \leq \epsilon^i \\ m_H, & \text{if } \left|\nu^i[k]\right| > \epsilon^i \end{cases},$$

where $\epsilon \in \mathbb{R}^N_{++}$ is a vector of thresholds. Since R is strategic, it chooses whether to incorporate the innovations using the signaling game strategy $\sigma_R(a \mid m)$, given the vector of messages m.

Define a *strategic innovation filter* by $D_{\sigma_R} : \mathbb{R}^N \to \mathbb{R}^N$ such that, given innovation ν, the components of gated innovation $\bar{\nu} = D_{\sigma_R}(\nu)$ are given by

$$\bar{\nu}^i = \begin{cases} \nu^i, & \text{if } a^i = a_T \\ 0, & \text{otherwise} \end{cases},$$

for $i \in \mathbb{S}$. Now we incorporate the function D_{σ_R} into the estimator by

$$\hat{x}[k+1] = A\hat{x}[k] + Bu[k] + L[k]D_{\sigma_R}\left(\nu[k]\right).$$

Feedback Controller

The optimal controller is given by the feedback law $u[k] = -K[k]\hat{x}[k]$, with gain

$$K[k] = \left(B'[k]S[K+1]B + R\right)^{-1} B'S[k+1]A,$$

where $S[k]$ is obtained by the backward Riccati difference equation

$$S[k] = A'\Big(S[k+1] - S[k+1]B \left(B'S[k+1]B + R\right)^{-1} B'S[k+1]\Big)A + Q,$$

with $S[T] = F$.

Control Criterion to Utility Mapping
The control cost J determines the signaling game utility of the device R. This utility should be monotonically decreasing in J. We consider a mapping $J \mapsto u_R^S$ defined by $u_R^S(\theta, m, a) = (\bar{v}_R - \underline{v}_R)e^{-\beta_R J} + \bar{v}_R$, where $\bar{v}_R$ and $\underline{v}_R$ denote maximum and minimum values of the utility, and β_R represents the sensitivity of the utility to the control cost.

8.3.4 Definition of Gestalt Nash Equilibrium

We now define the equilibrium concept for the overall game, which is called GNE. To differentiate this equilibrium from the equilibria in the `FlipIt` game and the signaling game, we use a superscript †.

Definition 8.3 (*Gestalt Nash equilibrium*) The triple $\left(p_A^\dagger, v_A^\dagger, v_D^\dagger\right)$, where $p_A^\dagger$ represents the probability of compromise of each of the cloud services, and $v_A^\dagger$ and $v_D^\dagger$ represent the vectors of equilibrium utilities for A^i and D^i, $i \in \mathbb{S}$, constitutes a GNE of the overall game if both Eqs. (8.20) and (8.21) are satisfied:

$$\forall i \in \{1, \ldots, m\}, \; p_A^{i\dagger} = T^{F_i}\left(v_A^{i\dagger}, v_D^{i\dagger}\right), \tag{8.20}$$

$$\left(\begin{bmatrix} v_A^{1\dagger} \\ v_A^{2\dagger} \\ \vdots \\ v_A^{N\dagger} \end{bmatrix}, \begin{bmatrix} v_D^{1\dagger} \\ v_D^{2\dagger} \\ \vdots \\ v_D^{N\dagger} \end{bmatrix}\right) \in T^S\left(\begin{bmatrix} p_A^{1\dagger} \\ p_A^{2\dagger} \\ \vdots \\ p_A^{N\dagger} \end{bmatrix}\right). \tag{8.21}$$

According to Definition 8.3, the overall game is at equilibrium when, simultaneously, each of the `FlipIt` games is at equilibrium and the one signaling game is at equilibrium.

8.4 Equilibrium Analysis

In this section, we give conditions under which a GNE exists. We start with a set of natural assumptions. Then we narrow the search for feasible equilibria. We show that the signaling game only supports pooling equilibria, and that only low-risk pooling equilibria survive selection criteria. Finally, we create a mapping that composes the signaling and `FlipIt` game models. We show that this mapping has a closed graph, and we use Kakutani's fixed-point theorem to prove the existence of a GNE. In order to avoid obstructing the flow of the chapter, we briefly summarize the proofs of each

Table 8.2 Assumptions

#	Assumption ($\forall i \in \mathbb{S}$)
A1	$0 = u_A^{S_i}(m_L, a_N) = u_A^{S_i}(m_H, a_N)$ $= u_D^{S_i}(m_L, a_N) = u_D^{S_i}(m_H, a_N)$
A2	$0 < u_A^{S_i}(m_L, a_T) < u_D^{S_i}(m_H, a_T)$ $< u_D^{S_i}(m_L, a_T) < u_A^{S_i}(m_H, a_T)$
A3	$\forall \theta^{-i}, m^{-i}, a^{-i}, \quad u_R^S(\theta, m, \bar{a}) > u_R(\theta, m, \tilde{a})$, where $\theta^i = \theta_A$, $m^i = m_H$, $\bar{a}^{-i} = \tilde{a}^{-i} = a^{-i}$, $\bar{a}^i = a_N$, and $\tilde{a}^i = a_T$
A4	$\forall \theta^{-i}, m^{-i}, a^{-i}, \quad u_R^S(\theta, m, \bar{a}) < u_R^S(\theta, m, \tilde{a})$, where $\theta^i = \theta_D$, $m^i = m_L$, $\bar{a}^{-i} = \tilde{a}^{-i} = a^{-i}$, $\bar{a}^i = a_N$, and $\tilde{a}^i = a_T$
A5	$\forall \theta, m^{-i}, a^{-i}, \quad u_R^S(\theta, \bar{m}, a) > u_R^S(\theta, \tilde{m}, a)$, where $a^i = a_T$, $\bar{m}^{-i} = \tilde{m}^{-i} = m^{-i}$, $\bar{m}^i = m_L$, and $\tilde{m}^i = m_H$

lemma, and we refer readers to the GNE derivations for a single cloud service in [21, 136].

8.4.1 Assumptions

For simplicity, let the utility functions of each signaling game sender i be dependent only on the messages and actions on cloud service i. That is, $\forall i \in \mathbb{S}$, $u_A^{S_i}(m, a) \equiv u_A^{S_i}(m^i, a^i)$ and $u_D^{S_i}(m, a) \equiv u_D^{S_i}(m^i, a^i)$. This can be removed, but it makes analysis more straightforward. Table 8.2 gives five additional assumptions. Assumption A1 assumes that each A^i and D^i, $i \in \mathbb{S}$, get zero utility when their messages are not trusted. A2 assumes an ordering among the utility functions for the senders in the signaling game. It implies that (a) A^i and D^i get positive utility when their messages are trusted; (b) for trusted messages, A prefers m_H to m_L; and (c) for trusted messages, D prefers m_L to m_H. These assumptions are justified if the goal of the attacker is to cause damage (with a high-risk message), while the defender is able to operate under normal conditions (with a low-risk message).

Assumptions A3–A4 give natural requirements on the utility function of the device. First, the worst-case utility for R is trusting a high-risk message from an attacker. Assume that, on every channel $i \in \mathbb{S}$, regardless of the messages and actions on the other channels, R prefers to play $a^i = a_N$ if $m^i = m_H$ and $\theta^i = \theta_A$. This is given by A3. Second, the best case utility for R is trusting a low-risk message from a defender. Assume that, on every channel $i \in \mathbb{S}$, regardless of the messages and actions on the other channels, R prefers to play $a^i = a_T$ if $m^i = m_L$ and $\theta^i = \theta_D$. This is given by A4. Finally, under normal operating conditions, R prefers trusted low-risk messages compared to trusted high-risk messages from both an attacker and a defender. This is given by A5.

8.4.2 GNE Existence Proof

We prove the existence of a GNE using Lemmas 8.4–8.10 and Theorem 8.11.

Narrowing the Search for GNE

Lemma 8.4 eliminates some candidates for GNE.

Lemma 8.4 (GNE Existence Regimes [21]) *Every GNE* $(p_A^\dagger, v_A^\dagger, v_D^\dagger)$ *satisfies:* $\forall i \in \mathbb{S}$, $v_A^i, v_D^i > 0$.

The basic idea behind the proof of Lemma 8.4 is that $v_A^i = 0$ or $v_D^i = 0$ cause either A^i or D^i to give up on capturing or recapturing the cloud. The cloud becomes either completely secure or completely insecure, neither of which can result in a GNE. Lemma 8.4 has a significant intuitive interpretation given by Remark 8.5.

Remark 8.5 In any GNE, for all $i \in \mathbb{S}$, R plays $a^i = a_T$ with nonzero probability. In other words, R never completely ignores any cloud service.

Elimination of Separating Equilibria

In signaling games, equilibria in which different types of senders transmit the same message are called *pooling equilibria*, while equilibria in which different types of senders transmit distinct messages are called *separating equilibria* [90]. The distinct messages in separating equilibria completely reveal the type of the sender to the receiver. Lemma 8.6 is typical of signaling games between players with opposed incentives.

Lemma 8.6 (No Separating Equilibria [21]) *For the signaling game, consider all pure strategy equilibria* $\left(\sigma_R^*; \sigma_A^{1*}, \ldots \sigma_A^{N*}; \sigma_D^{1*}, \ldots \sigma_D^{N*}\right)$ *in which each* A^i *and* D^i, $i \in \mathbb{S}$, *receive positive expected utility. All such equilibria satisfy* $\sigma_A^{i*}(m) = \sigma_D^{i*}(m)$ *for all* $m \in M$ *and* $i \in \mathbb{S}$. *That is, the senders on each cloud service* i *use pooling strategies.*

Lemma 8.6 holds because it is never incentive-compatible for an attacker A^i to reveal his type, in which case R would not trust A^i. Hence, A^i always imitates D^i by pooling.

Signaling Game Equilibrium Selection Criteria

Four pooling equilibria are possible in the signaling game: A^i and D^i transmit $m^i = m_L$ and R plays $a^i = a_T$ (which we label **EQ-L1**), A^i and D^i transmit $m^i = m_L$ and R plays $a^i = a_N$ (**EQ-L2**), A^i and D^i transmit $m^i = m_H$ and R plays $a^i = a_T$ (**EQ-H1**), and A^i and D^i transmit $m^i = m_H$ and R plays $a^i = a_N$ (which we label **EQ-H2**). In fact, the signaling game always admits multiple equilibria. Lemma 8.7 performs equilibrium selection.

Lemma 8.7 (Selected Equilibria) *The* intuitive criterion *[170] and the criterion of* first mover advantage *imply that equilibria* ***EQ-L1*** *and* ***EQ-L2*** *will be selected.*

Proof The first mover advantage states that, if both A^i and D^i prefer one equilibrium over the others, they will choose the preferred equilibrium. Thus, A^i and D^i will always choose **EQ-L1** or **EQ-H1** if either of those is admitted. When neither is admitted, we select **EQ-L2**.[3] When both are admitted, we use the intuitive criterion to select among them. Assumption A2 states that A^i prefers **EQ-H1**, while D^i prefers **EQ-L1**. Thus, if a sender deviates from **EQ-H1** to **EQ-L1**, R can infer that the sender is a defender, and trust the message. Therefore, the intuitive criterion rejects **EQ-H1** and selects **EQ-L1**. Finally, Assumption A5 can be used to show that **EQ-H1** is never supported without **EQ-L1**. Hence, only **EQ-L1** and **EQ-L2** survive the selection criteria.

At the boundary between the parameter regime that supports **EQ-L1** and the parameter regime that supports **EQ-L2**, R can choose any mixed strategy, in which he plays both $a^i = a_T$ and $a^i = a_N$ with some probability. Indeed, for any cloud service $i \in \mathbb{S}$, hold p_A^j, $j \neq i$ and $j \in \mathbb{S}$, constant, and let $p_A^{i\diamond}$ denote the boundary between the **EQ-L1** and **EQ-L2** regions. Then Remark 8.8 gives an important property of $p_A^{i\diamond}$.

Remark 8.8 By Lemma 8.4, all GNE satisfy $p_A^i \leq p_A^{i\diamond}$. Therefore, $p_A^{i\diamond}$ is a worst-case probability of compromise.

Remark 8.8 is a result of the combination of the signaling and `FlipIt` games. Intuitively, it states that strategic trust in the communication layer is able to limit the probability of compromise of a cloud service, regardless of the attack and defense costs in the cyber layer.

`FlipIt` Game Properties

For the `FlipIt` games on each cloud service $i \in \mathbb{S}$, denote the ratio of attacker and defender expected utilities by $v_A^i = v_A^i / v_D^i$. For $i \in \mathbb{S}$, define the set $\mathbb{V}^i$ by

$$\mathbb{V}^i = \left\{ v \in \mathbb{R}_+ \,:\, 0 \leq v \leq u_A^{S_i}(m_L, a_T) / u_D^{S_i}(m_L, a_T) \right\}.$$

Also define the set $\mathbb{PR}^i$, $i \in \mathbb{S}$, by $\mathbb{PR}^i =$

$$\left\{ p \in [0,1] \,:\, 0 < p < T^{F_i}\left(u_A^{S_i}(m_L, a_T), u_D^{S_i}(m_L, a_T) \right) \right\}.$$

Next, for $i \in \mathbb{S}$, define modified `FlipIt` game mappings $\tilde{T}^{F_i} : \mathbb{V}^i \to \mathbb{PR}^i$, where

$$p_A^{i*} = \tilde{T}^{F_i}\left(v_A^i\right) \iff p_A^{i*} \in T^{F_i}\left(v_A^i, v_D^i\right). \tag{8.22}$$

Then Lemma 8.9 holds.

Lemma 8.9 (Continuity of $\tilde{T}^{F_i}$ [136]) *For $i \in \mathbb{S}$, $\tilde{T}^{F_i}(v_A^i)$ is continuous in $v_A^i \in \mathbb{V}^i$.*

[3] This is without loss of generality, since A1 implies that the sender utilities are the same for **EQ-L2** and **EQ-H2**.

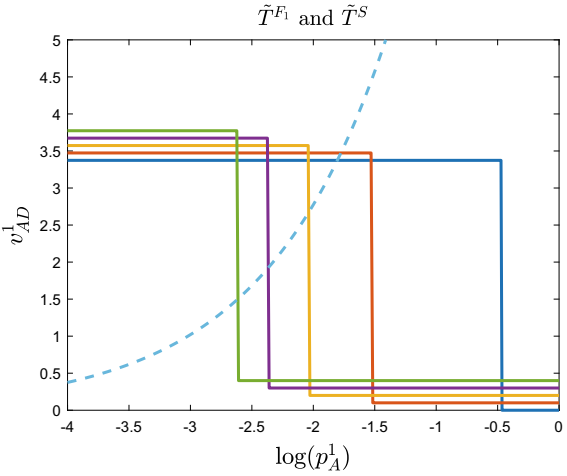


Fig. 8.5 The (solid) step functions depict modified signaling game mappings $\tilde{T}^S$ for five different sets of parameters. The (dashed) curve depicts a modified `FlipIt` game mapping $\tilde{T}^{F_1}$. The intersection is a GNE. The figure shows only one dimension out of N dimensions

The dashed curve in Fig. 8.5 gives an example of $\tilde{T}^{F_i}$ for $i = 1$. The independent variable is on the vertical axis, and the dependent variable is on the horizontal axis.

Signaling Game Properties

Let $v_A = [v_A^i]_{i \in \mathbb{S}}$, $\mathbb{V} = \prod_{i \in \mathbb{S}} \mathbb{V}^i$, and $\mathbb{PR} = \prod_{i \in \mathbb{S}} \mathbb{PR}^i$. Define a modified signaling game mapping by $\tilde{T}^S : \mathbb{PR} \to \mathcal{P}(\mathbb{V})$ such that

$$v_A^* \in \tilde{T}^S(p_A) \iff \left(v_A^*, v_D^*\right) \in T^S(p_A), \tag{8.23}$$

where T^S selects the equilibria given by Lemma 8.7. Then we have Lemma 8.10.

Lemma 8.10 (Properties of $\tilde{T}^S$) *Construct a graph*

$$\mathbb{G} = \left\{ \left(p_A, v_A^*\right) \in \mathbb{PR} \times \mathbb{V} : v_A^* \in \tilde{T}^S(p_A) \right\},$$

The graph $\mathbb{G}$ is closed. Additionally, for every $p_A \in \mathbb{PR}$, the set of outputs of $\tilde{T}^S(p_A)$ is non-empty and convex.

Proof The graph $\mathbb{G}$ is closed because it contains all of its limit points. The set of outputs is non-empty because a signaling game equilibrium exists for all p_A. It is convex because expected utilities for mixed-strategy equilibria are convex combinations of pure strategy utilities and because assumption A2 implies that convexity also holds for the ratio of the utilities.

The step functions (plotted with solid lines) in Fig. 8.5 plot example mappings from p_A^1 on the horizontal axis to v_A^1 on the vertical axis for $v_A \in \tilde{T}^S(p_A)$, holding p_A^i, $i \in \{2, 3, \ldots, N\}$ fixed. It is clear that the graphs are closed.

Fixed-Point Theorem
By combining Eqs. (8.20)–(8.21) with Eqs. (8.22) and (8.23), we see that the vector of equilibrium utility ratios $v_A^\dagger = [v_A^{i\dagger}]_{i\in\mathbb{S}}$ in any GNE $(p_A^\dagger, v_A^\dagger, v_D^\dagger)$ must satisfy

$$\begin{bmatrix} v_A^{1\dagger} \\ v_A^{2\dagger} \\ \vdots \\ v_A^{N\dagger} \end{bmatrix} \in \tilde{T}^S \left(\begin{bmatrix} \tilde{T}^{F_1}\left(v_A^{1\dagger}\right) \\ \tilde{T}^{F_2}\left(v_A^{2\dagger}\right) \\ \vdots \\ \tilde{T}^{F_2}\left(v_A^{N\dagger}\right) \end{bmatrix} \right).$$

Denote this composed mapping by $\tilde{T}^{S\circ F} : \mathbb{V} \to \mathcal{P}(\mathbb{V})$ such that the GNE requirement can be written by $v_A^\dagger \in \tilde{T}^{S\circ F}(v_A^\dagger)$. Figure 8.5 gives a one-dimensional intuition behind Theorem 8.12. The example signaling game step functions $\tilde{T}^S$ have closed graphs and the outputs of the functions are non-empty and convex. The `FlipIt` curve $\tilde{T}^{F_1}$ is continuous. The two mappings are guaranteed to intersect and the intersection is a GNE.

According to Lemma 8.10, the graph $\mathbb{G}$ of the signaling game mapping is closed, and the set of outputs of $\tilde{T}^S$ is non-empty and convex. Since each modified `FlipIt` game mapping $\tilde{T}^{F_i}$, $i \in \mathbb{S}$ is a continuous function, each $\tilde{T}^{F_i}$ produces a closed graph and has non-empty and (trivially) convex outputs. Thus, the graph of the composed mapping, $\tilde{T}^{S\circ F}$, is also closed and has non-empty and convex outputs. Because of this, we can apply Kakutani's fixed-point theorem, famous for its use in proving Nash equilibrium.

Theorem 8.11 (Kakutani Fixed-Point Theorem [171])—*Let Φ be a non-empty, compact, and convex subset of some Euclidean space $\mathbb{R}^n$. Let $Z : \Phi \to \mathcal{P}(\Phi)$ be a set-valued function on Φ with a closed graph and the property that, for all $\phi \in \Phi$, $Z(\phi)$ is non-empty and convex. Then Z has a fixed point.*

The mapping $\tilde{T}^{S\circ F}$ is a set-valued function on $\mathbb{V}$, which is a non-empty, compact, and convex subset of $\mathbb{R}^N$. $\tilde{T}^{S\circ F}$ also has a closed graph, and the set of its outputs is non-empty and convex. Therefore, $\tilde{T}^{S\circ F}$ has a fixed-point, which is precisely the definition of a GNE. Hence, we have Theorem 8.12.

Theorem 8.12 (GNE Existence) *Let the utility functions in the signaling game satisfy Assumptions A1–A5. Then a GNE exists.*

Proof The proof has been constructed from Lemmas 8.4–8.10 and Theorem 8.11.

8.4.3 Adaptive Algorithm

Numerical simulations suggest that Assumptions A1–A5 often hold. If this is not the case, however, Algorithm 8.1 can be used to compute the GNE. The main idea of the

Algorithm 8.1 Adaptive defense algorithm for iSTRICT

1. Initialize parameters α_A^i, α_D^i, p_A^i, p_D^i, $\forall i \in \mathbb{S}$, in each `FlipIt` game, and σ_A^i and σ_D^i, $\forall i \in \mathbb{S}$, σ_R in the signaling game
   Signaling game:
2. Solve optimization problems in Eqs. (8.11) and (8.12), respectively, and obtain σ_A^{i*} and σ_D^{i*}, $\forall i \in \mathbb{S}$
3. Update belief $\mu^i(\theta_A \mid m^i)$ based on Eq. (8.14), and $\mu^i(\theta_D \mid m^i) = 1 - \mu^i(\theta_A \mid m^i)$, $\forall i \in \mathbb{S}$
4. Solve receiver's problem in Eq. (8.13) and obtain σ_R^*
5. If σ_A^{i*}, σ_D^{i*}, σ_R^* do not change, go to step 6; otherwise, go back to step 2
6. Obtain v_A^i and v_D^i, $\forall i \in \mathbb{S}$, from Eqs. (8.15) and (8.16), respectively
   `FlipIt` **game:**
7. Solve defenders' and attackers' problems in Eqs. (8.6) and (8.7) jointly, and obtain f_A^{i*} and f_D^{i*}, $\forall i \in \mathbb{S}$
8. Map the frequency pair (f_A^{i*}, f_D^{i*}) to the probability pair (p_A^{i*}, p_D^{i*}) through Eq. (8.3), $\forall i \in \mathbb{S}$
9. If (p_D^{i*}, p_A^{i*}), $\forall i \in \mathbb{S}$, do not change, go to step 10; otherwise, go back to step 2
10. **Return** $p_A^\dagger := p_A^*$, $\sigma_A^{i\dagger} := \sigma_A^{i*}$, $\sigma_D^{i\dagger} := \sigma_D^{i*}$, $\forall i \in \mathbb{S}$, and $\sigma_R^\dagger := \sigma_R^*$

adaptive algorithm is to update the strategic decision-making of different entities in iSTRICT iteratively.

Given the probability vector p_A, Lines 2–5 of Algorithm 8.1 compute a PBNE for the signaling game which consists of the strategy profile

$$\left(\sigma_R^*; \sigma_A^{1*}, \ldots \sigma_A^{N*}; \sigma_D^{1*}, \ldots \sigma_D^{N*}\right)$$

and belief vector $\mu(\theta \mid m)$. The algorithm computes the PBNE iteratively using best response. The vectors of equilibrium utilities $\left(v_A^*, v_D^*\right)$ are given by Eqs. (8.15) and (8.16). Using $\left(v_A^*, v_D^*\right)$, Line 7 of Algorithm 8.1 updates the equilibrium strategies of the `FlipIt` games and arrives at a new prior probability pair (p_A^{i*}, p_D^{i*}), $\forall i \in \mathbb{S}$, through the mapping in Eq. (8.3). This initializes the next round of the signaling game with the new (p_A^{i*}, p_D^{i*}). The algorithm terminates when the probabilities remain unchanged between rounds.

To illustrate Algorithm 8.1, we next present an example including $N = 4$ cloud services. The detailed physical meaning of each service will be presented in Sect. 8.5. Specifically, the costs of renewing control of cloud services are $\alpha_A^1 = \$2\text{k}$, $\alpha_A^2 = \$0.8\text{k}$, $\alpha_A^3 = \$10\text{k}$, $\alpha_A^4 = \$12\text{k}$, and $\alpha_D^1 = \$0.2\text{k}$, $\alpha_D^1 = \$0.1\text{k}$, $\alpha_D^1 = \$0.05\text{k}$, $\alpha_D^1 = \$0.03\text{k}$, for the attackers and defenders, respectively. The initial proportions of time of each attacker and defender controlling the cloud services are $p_A^1 = 0.2$, $p_A^2 = 0.4$, $p_A^3 = 0.6$, $p_A^4 = 0.15$, and $p_D^1 = 0.8$, $p_D^2 = 0.6$, $p_D^3 = 0.4$, $p_D^4 = 0.85$, respectively. For the signaling game in the communication layer, the initial probabilities that attacker sends low-risk message at each cloud service are equal to 0.2, 0.3, 0.1, and 0.4, respectively. Similarly, the defender's initial probabilities of sending low-risk message are equal to 0.9, 0.8, 0.95, and 0.97, respectively.

Figures 8.6, 8.7 and 8.8 present the results of Algorithm 8.1 on this example system. The result in Fig. 8.8 shows that cloud services 1 and 2 can be compromised

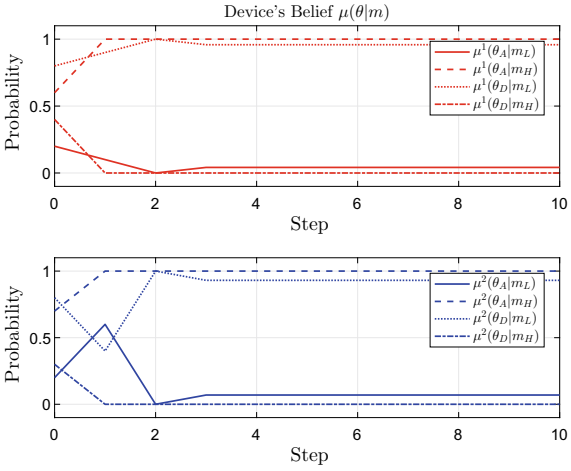


Fig. 8.6 Evolution of the device belief for the adaptive algorithm with four cloud services. The algorithm converges in four steps

by the attacker. Figure 8.6 shows the device's belief on the received information. At the GNE, the attacker and defender both send low-risk messages. Four representative devices' actions are shown in Fig. 8.7. The devices strategically reject low-risk message in some cases due to the couplings between layers in iSTRICT. Because of the large attack and defense cost ratios and the crucial impact on physical system performance of services 3 and 4, $p_D^3 = p_D^4 = 1$, insuring a secure information provision. In addition, the defense strategies at the cloud layer and the communication layer are adjusted adaptively according to the attackers' behaviors. Within each layer, all players are required to best respond to the strategies of the other players. This cross-layer approach enables a defense-in-depth mechanism for the devices in iSTRICT.

8.5 Application to Autonomous Vehicle Control

In this section, we apply iSTRICT to a cloud-enabled autonomous vehicle network. Two autonomous vehicles use an observer to estimate their positions and velocities based on measurements from six sources, four of which may be compromised. They also implement optimal feedback control based on the estimated state.

8.5.1 Autonomous Vehicle Security

Autonomous vehicle technology will make a powerful impact on several industries. In the automotive industry, traditional car companies, as well as technology firms such as Google [172] are racing to develop autonomous vehicle technology. Maritime

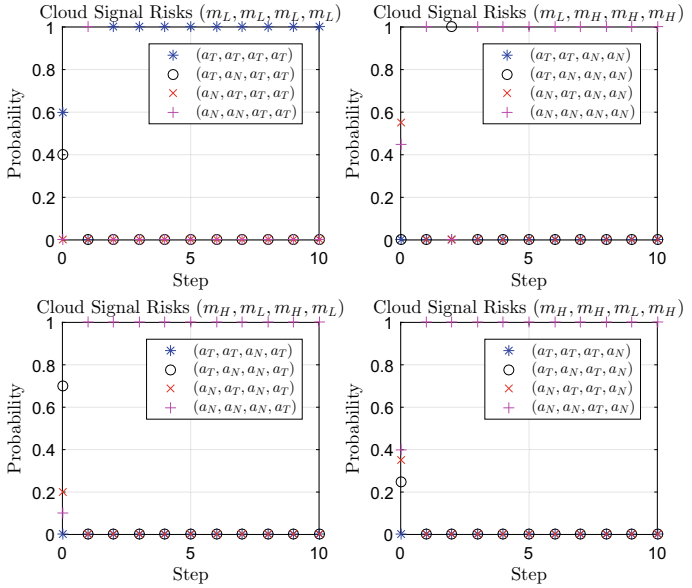


Fig. 8.7 Evolution of the device action for the same scenario as in Fig. 8.6

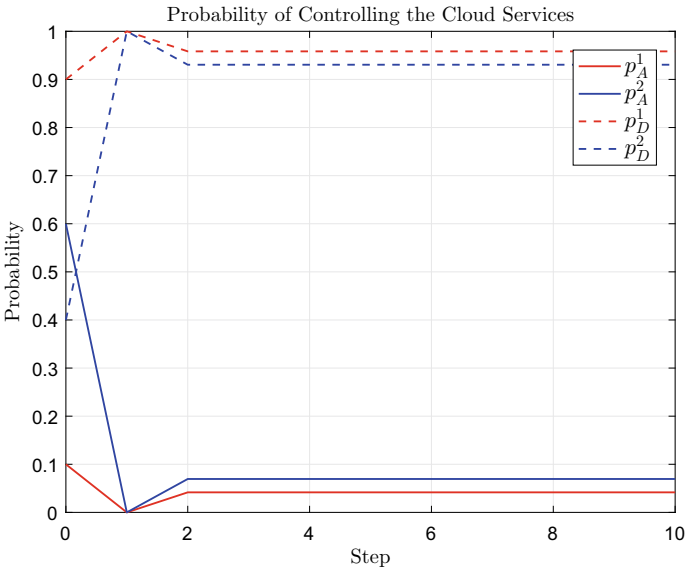


Fig. 8.8 Evolution of the `FlipIt` game equilibrium for the same scenario as in Fig. 8.6

shipping is also an attractive application of autonomous vehicles. Autonomous ships are expected to be safer, higher-capacity, and more resistant to piracy attacks [173]. Finally, unmanned aerial vehicles (UAVs) have the potential to reshape fields such as mining, disaster relief, and precision agriculture [174].

Nevertheless, autonomous vehicles pose clear safety risks. In ground transportation, in March of 2018, an Uber self-driving automobile struck and killed a pedestrian [175]. On the sea, multiple crashes of ships in the United States Navy during 2017 [176], have prompted concerns about too much reliance on automation. In the air, cloud-enabled UAVs could be subject to data integrity or availability attacks [177]. In general, autonomous vehicles rely on many remote sources (e.g., other vehicles, GPS signals, location-based services) for information. In the most basic case, these sources are subject to errors that must be handled robustly. In addition, the sources could also be selfish and strategic. For instance, an autonomous ship could transmit its own coordinates dishonestly in order to clear its own shipping path of other vessels. In the worst-case, the sources could be malicious. An attacker could use a spoofed GPS signal in order to destroy a UAV or to use the UAV to attack another target. In all of these cases, autonomous vehicles must decide whether to trust the remote sources of information.

8.5.2 *Physical-Layer Implementation*

We consider an interaction between nine agents. Two autonomous vehicles implement observer-based optimal feedback control according to the iSTRICT framework. Each vehicle has two states: position and angle. Thus, the combined system has the state vector $x[k] \in \mathbb{R}^4$ described in Fig. 8.9. The states evolve over finite horizon $k \in \{0, 1, \ldots, T\}$.

These states are observed through both remote and local measurements. Figure 8.10 describes these measurements. The local measurements $y^5[k]$ and $y^6[k]$ originate from sensors on the autonomous vehicle, so these are secure. Hence, the autonomous vehicle always trusts $y^5[k]$ and $y^6[k]$. In addition, while the magnetic compass sensors are subject to electromagnetic attack, this involves high attack costs α_A^3 and α_A^4. The defense algorithm yields that R always trusts $y^3[k]$ and $y^4[k]$ at the GNE. The remote measurements $\tilde{y}^1[k]$ and $\tilde{y}^2[k]$ are received from cloud services that may be controlled by defenders D^1 and D^2, or that may be compromised by attackers A^1 and A^2.

In the signaling game, attackers A^1 and A^2 may add bias terms $\Delta_A^1[k]$ or $\Delta_A^2[k]$ if $\theta^1 = \theta_A$ or $\theta^2 = \theta_A$, respectively. Therefore, the autonomous vehicles must strategically decide whether to trust these measurements. Each $\tilde{y}^i[k]$, $i \in \{1, 2\}$, is classified as a low-risk ($m^i = m_L$) or high-risk ($m^i = m_H$) message according to an innovation filter. R decides whether to trust each message according to the action vector $a = [\, a^1 \; a^2 \,]'$, where $a^1, a^2 \in \{a_N, a_T\}$. We seek an equilibrium of the signaling game that satisfies Definition 8.2.

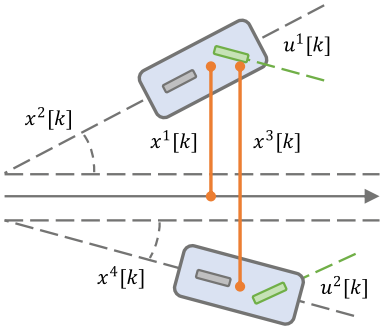


Fig. 8.9 We use bicycle steering models from [178] to conceptually capture the vehicle dynamics. The vehicle states are given by $x^1[k]$: first vehicle position, $x^2[k]$: first vehicle angle; $x^3[k]$: offset between vehicles; $x^4[k]$: second vehicle angle. Controls $u^1[k]$ and $u^2[k]$ represent the steering angles of the first and second vehicles, respectively

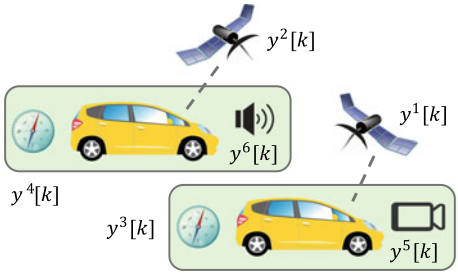
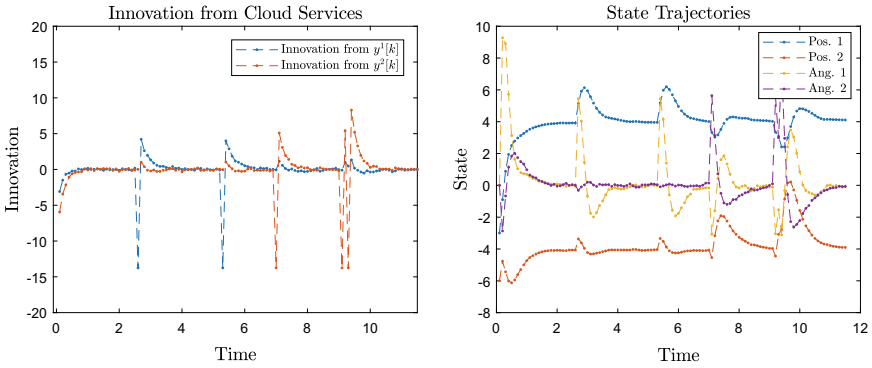


Fig. 8.10 Local sensors include a localization camera on vehicle 1 and a range finding device on vehicle 2. Remote sensors include magnetic compass sensors and GPS receivers on both vehicles

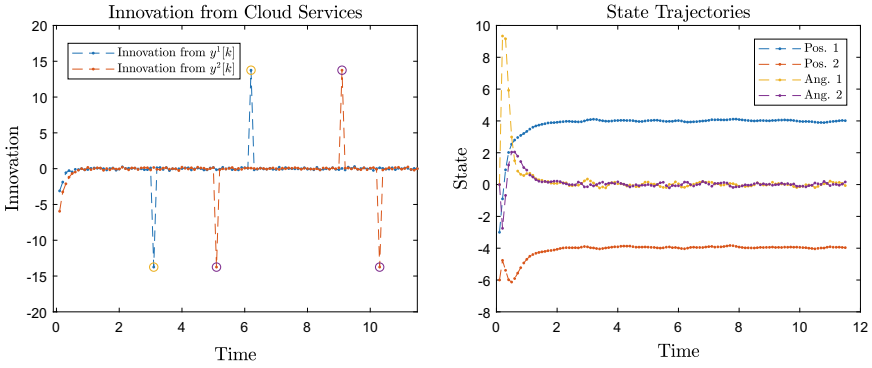
8.5.3 Signaling Game Results

Figure 8.11 depicts the results of three different signaling-game strategy profiles for the attackers, defenders, and device, using the software MATLAB [179]. The observer and controller are linear, so the computation is rapid. Each iteration of the computational elements of the control loop depicted in Fig. 8.4 takes less than 0.0002 s on a Lenovo ThinkPad L560 laptop with 2.30 GHz processor and 8.0 GB of installed RAM.

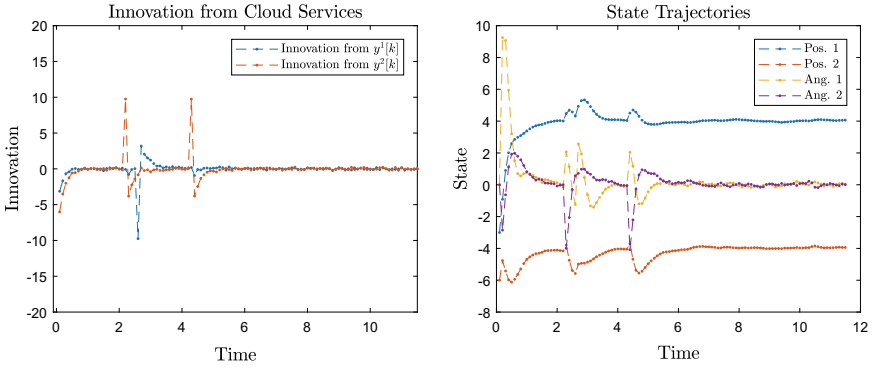
In all of the scenarios, we set the position of the first vehicle to track a reference trajectory of $x^1[k] = 4$, the offset between the second vehicle and the first vehicle to track a reference trajectory of $x^3[k] = 8$, and both angles to target $x^2[k] = x^4[k] = 0$. Row 1 depicts a scenario in which A^1 and A^2 send m_H and R plays $a^1 = a^2 = a_T$. The spikes in the innovation represent the bias terms added by the attacker when he controls the cloud. The spikes in Fig. 8.11a are large because the attacker adds bias terms corresponding to high-risk messages. These bias terms cause large deviations in the position and angle from their desired values (Fig. 8.11b). For instance, at time 10, the two vehicles come within approximately 4 units of each other.



(a) Innovation (b) State trajectories



(c) Innovation (d) State trajectories



(e) Innovation (f) State trajectories

Fig. 8.11 Row 1: A^1 and A^2 send m_H and R plays $[\,a_T\ a_T\,]'$, Row 2: A^1 and A^2 send m_H and R plays $[\,a_N\ a_N\,]'$, Row 3: A^1 and A^2 send m_L and R plays $[\,a_T\ a_T\,]'$. Column 1: innovation, Column 2: state trajectories

Row 2 depicts the best response of R to this strategy. The vehicle uses an innovation filter (here, at $\epsilon^1 = \epsilon^2 = 10$) which categorizes the biased innovations as m_H. The best response is to choose

$$\sigma_R\left(\begin{bmatrix} a_T \\ a_T \end{bmatrix} | \begin{bmatrix} m_L \\ m_L \end{bmatrix}\right) = \sigma_R\left(\begin{bmatrix} a_T \\ a_N \end{bmatrix} | \begin{bmatrix} m_L \\ m_H \end{bmatrix}\right) = 1,$$

$$\sigma_R\left(\begin{bmatrix} a_N \\ a_T \end{bmatrix} | \begin{bmatrix} m_H \\ m_L \end{bmatrix}\right) = \sigma_R\left(\begin{bmatrix} a_N \\ a_N \end{bmatrix} | \begin{bmatrix} m_H \\ m_H \end{bmatrix}\right) = 1,$$

i.e., to trust only low-risk messages. The circled data points in Fig. 8.11c denote high-risk innovations from the attacker that are rejected. Figure 8.11d shows that this produces very good results in which the positions of the first and second vehicle converge to their desired values of 4 and -4, respectively, and the angles converge to 0.

But iSTRICT assumes that the attackers are also strategic. A^1 and A^2 realize that high-risk messages will be rejected, so they add smaller bias terms $\Delta_A^1[k]$ and $\Delta_A^2[k]$ which are classified as m_L. This is depicted by Fig. 8.11e. It is not optimal for the autonomous vehicle to reject all low-risk messages, because most such messages come from a cloud controlled by the defender. Therefore, the device must play $a^1 = a^2 = a_T$. Nevertheless, Fig. 8.11f shows that these low-risk messages create significantly less disturbance than the disturbances from high-risk messages in Fig. 8.11b. In summary, the signaling-game equilibrium is for A^1, A^2, D^1, and D^2 to transmit low-risk messages and for R to trust low-risk messages while rejecting high-risk messages off the equilibrium path.

8.5.4 *Results of the* `FlipIt` *Games*

Meanwhile, A^1 and D^1 play a `FlipIt` game for control of Cloud Service 1, and A^2 and D^2 play a `FlipIt` game for control of Cloud Service 2. Based on the equilibrium of the signaling game, all players realize that the winners of the `FlipIt` games will be able to send trusted low-risk messages, but not trusted high-risk messages. Based on Assumption A2, low-risk messages are more beneficial to the defenders than to the attackers. Hence, the incentives to control the cloud are larger for defenders than for attackers. This results in a low p_A^1 and p_A^2 from the `FlipIt` game. If the equilibrium from the previous subsection holds for these prior probabilities, then the overall five-player interaction is at a GNE as described in Definition 8.3 and Theorem 8.12.

Table 8.3 is useful for benchmarking the performance of iSTRICT. The table lists the empirical value of the control criterion given by Eq. (8.19) for Fig. 8.11. Column 1 is the benchmark case, in which A^1 and A^2 add high-risk noise, and the noise is mitigated somewhat by a Kalman filter, but the bias is not handled optimally. Column 2 shows the improvement provided by iSTRICT against a nonstrategic attacker, and

Table 8.3 Control costs for the simulations depicted in Fig. 8.11

	Ungated m_H	Gated m_H	Trusted m_L
Trial 1	274,690	42,088	116,940
Trial 2	425,520	42,517	123,610
Trial 3	119,970	42,444	125,480
Trial 4	196,100	42,910	89,980
Trial 5	229,870	42,733	66,440
Trial 6	139,880	42,412	69,510
Trial 7	129,980	42,642	116,560
Trial 8	97,460	42,468	96,520
Trial 9	125,490	42,633	50,740
Trial 10	175,670	42,466	78,700
Average	191,463	42,531	93,448

Column 3 shows the improvement provided by iSTRICT against a strategic attacker. The improvement is largest in Column 2, but it is significant against a strategic attacker as well.

8.5.5 GNE for Different Parameters

Now consider a parameter change in which A^2 develops new malware to compromise the GPS position signal $\tilde{y}^2[k]$ at a much lower cost α_A^2. (See Sect. 8.3.1). In equilibrium, this increases p_A^2 from 0.03 to 0.10. A higher number of perturbed innovations are visible in Fig. 8.12a. This leads to the poor state trajectories of Fig. 8.12b. The control cost from Eq. (8.19) increases, and the two vehicles nearly collide at time 8. The large changes in angles show that the vehicles turn rapidly in different directions.

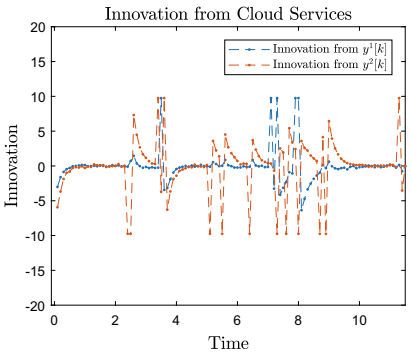
In this case, R's best response is

$$\sigma_R\left(\begin{bmatrix} a_T \\ a_N \end{bmatrix} | \begin{bmatrix} m_L \\ m_L \end{bmatrix}\right) = \sigma_R\left(\begin{bmatrix} a_T \\ a_N \end{bmatrix} | \begin{bmatrix} m_L \\ m_H \end{bmatrix}\right) = 1,$$

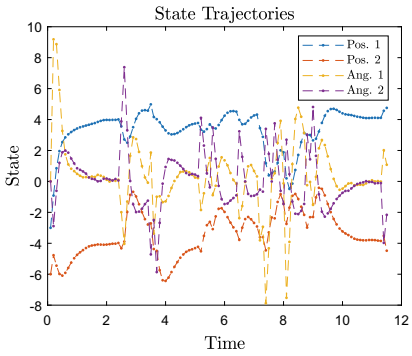
$$\sigma_R\left(\begin{bmatrix} a_N \\ a_N \end{bmatrix} | \begin{bmatrix} m_H \\ m_L \end{bmatrix}\right) = \sigma_R\left(\begin{bmatrix} a_N \\ a_N \end{bmatrix} | \begin{bmatrix} m_H \\ m_H \end{bmatrix}\right) = 1,$$

i.e., to not trust even the low-risk messages from the remote GPS signal. The circles on $\nu^2[k]$ for all k in Fig. 8.12c represent not trusting. The performance improvement can be seen in Fig. 8.12d.

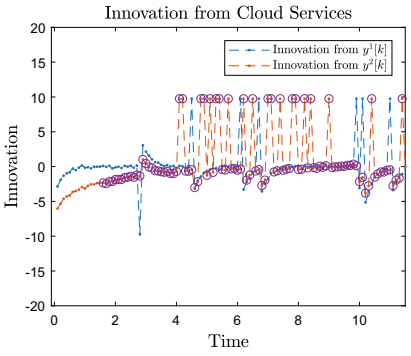
Interestingly, though, Remark 8.5 states that this cannot be an equilibrium. In the `FlipIt` game, A^2 would have no incentive to capture Cloud Service 2, since R never trusts that cloud service. This would lead to $p_A^2 = 0$. Moving forward, R



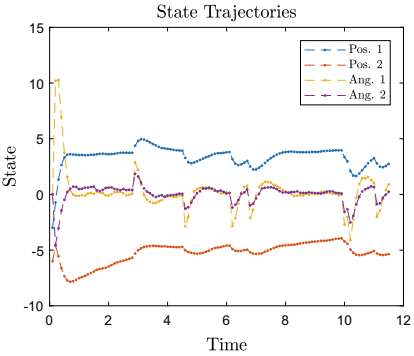
(a) Innovation



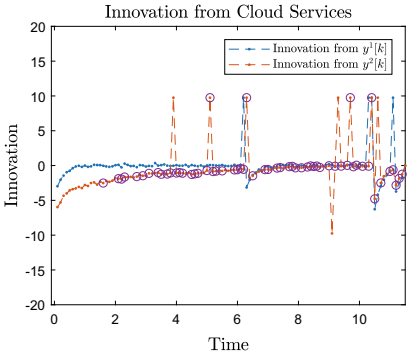
(b) State trajectories



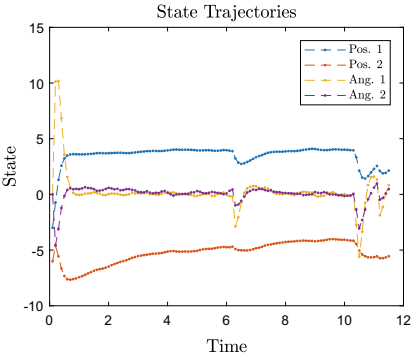
(c) Innovation



(d) State trajectories



(e) Innovation



(f) State trajectories

Fig. 8.12 Row 1: innovation and state trajectories for $p_A^2 = 0.10$ and R plays $[\, a_T \;\; a_T \,]'$, Row 2: innovation and state trajectories for $p_A^2 = 0.10$ and R plays $[\, a_T \;\; a_N \,]'$, Row 3: innovation and state trajectories in which R mixes strategies between $[\, a_T \;\; a_T \,]'$ and $[\, a_T \;\; a_N \,]'$

Table 8.4 Control costs for the simulations depicted in Fig. 8.12

	Trusted, frequent m_L	Untrusted, frequent m_L	Mixed trust with m_L
Trial 1	185,000	128,060	146,490
Trial 2	211,700	121,910	143,720
Trial 3	213,500	144,090	130,460
Trial 4	239,400	138,350	135,930
Trial 5	94,400	135,160	139,680
Trial 6	2,581,500	119,700	125,270
Trial 7	254,000	138,160	122,790
Trial 8	1,020,000	130,260	146,370
Trial 9	250,900	138,960	151,470
Trial 10	4,182,600	135,780	126,550
Average	923,300	133,043	136,873

would trust Cloud Service 2 in the next signaling game, and A^2 would renew his attacks. iSTRICT predicts that this pattern of compromising, not trusting, trusting, and compromising would repeat in a limit cycle, and not converge to equilibrium.

A mixed-strategy equilibrium, however, does exist. R chooses a mixed strategy in which he trusts low-risk messages on Cloud Service 2 with some probability. This probability incentivizes A^2 to attack the cloud with a frequency between those that best respond to either of R's pure strategies. At the GNE, the attack frequency of A^2 produces $0 < p_A^2 < 0.10$ in the `FlipIt` game. In fact, this is the worst-case $p_A^2 = p_A^{2\diamond}$ from Remark 8.8. In essence, R's mixed-strategy serves as a last-resort countermeasure to the parameter change due to the new malware obtained by A^2.

Figure 8.12e depicts the innovation with a mixed strategy in which R sometimes trusts Cloud Service 2. Figure 8.12f shows the impact on state trajectories. At this mixed-strategy equilibrium, A^1, A^2, D^1, and D^2 choose optimal attack/recapture frequencies in the cloud layer and send optimal messages in the communications layer, and R optimally chooses which messages to trust in the communication layer based on an innovation filter and observer-based optimal control in the physical layer. No players have incentives to deviate from their strategies at the GNE.

Table 8.4 quantifies the improvements provided by iSTRICT in these cases. Column 1 is the benchmark case, in which an innovation gate forces A^2 to add low-risk noise, but his frequent attacks still cause large damages. Column 2 gives the performance of iSTRICT against a strategic attacker, and Column 3 gives the performance of iSTRICT against a nonstrategic attacker. In both cases, the cost criterion decreases by a factor of at least six.

8.6 Discussion of Results

The model introduced in this chapter mitigates malicious deception through multiple, interdependent layers of defense, emphasizing techniques that account for the dynamic processes controlled by the IoCT. At the lowest physical layer, a Kalman filter handles sensor noise. The Kalman filter, however, is not designed for the large bias terms that can be injected into sensor measurements by attackers. We use an innovation gate in order to reject these large bias terms. But even measurements within the innovation gate should be rejected if there is a sufficiently high-risk that a cloud service is compromised. We determine this threshold risk level strategically, using a signaling game. Now, it may not be possible to estimate these risk levels using past data. Instead, iSTRICT estimates the risk proactively using `FlipIt` games. The equilibria of the `FlipIt` games depend on the incentives of the attackers and defenders to capture or reclaim the cloud. These incentives result from the outcome of the signaling game, which means that the equilibrium of the overall interaction consists of a fixed point between mappings that characterize the `FlipIt` games and the signaling game. This equilibrium is a GNE.

8.7 Related Work

The iSTRICT framework in this chapter builds on two existing game models. One is the signaling game which has been used in intrusion detection systems [59] and network defense [135]. The other one is the `FlipIt` game [158, 162] which has been applied to the security of a single cloud service [20, 136] as well as AND/OR combinations of cloud services [180]. In terms of the technical framework, iSTRICT builds on existing achievements in IoCT architecture design [7, 52, 181, 182], which describe the roles of different layers of the IoCT at which data is collected, processed, and accessed by devices [52]. Each layer of the IoCT consists of different enabling technologies such as wireless sensor networks and data management systems [182].

8.8 Notes

Designing trustworthy cloud service systems has been investigated extensively in the literature. Various methods, including a feedback evaluation component, Bayesian game, and domain partition have been proposed [167, 183, 184]. Trust models to predict the cloud trust values (or reputation) can be mainly divided into objective and subjective classes. The first are based on the quality of service parameters, and the second are based on feedback from cloud service users [167, 185]. In the IoCT, agents may not have the sufficient number of interactions, which makes reputation challenging to obtain [46].

Chapter 9
Active Crowd Defense

Defenders in Chap. 8 countered malicious deception by strategically deciding whether to trust other agents from whom they received commands. This is a logical foundation for the mitigation of deception. In some situations, however, this passive approach is insufficient. Attackers may use "broadcast deception," in which malware scans a large number of nodes in search of easy targets, such as nodes in a network that use the default username and password settings. Even finding a small number of easy targets is sufficient for the attackers, since nodes with stronger defenses merely stop the attack but do not punish the attacker. In this chapter, we develop a mechanism for *active defense* in which strong nodes attempt to report and shut down detected attackers.

This active defense is used to combat an emerging type of attack called a *"physical" denial-of-service* (*PDoS*) *attack*, in which IoCT devices overflow the "physical bandwidth" of a CPS. The chapter quantifies the population-based risk to a group of IoCT devices targeted by malware for a PDoS attack. In order to model active defense against a PDoS attack, we develop a "Poisson signaling game," a signaling game with an unknown number of receivers, which have varying abilities to detect deception. Then we use a version of this game to analyze two mechanisms to deter botnet recruitment. Equilibrium results indicate that (1) defenders can bound botnet activity, and (2) legislating a minimum level of security has only a limited effect, while incentivizing active defense can decrease botnet activity arbitrarily.

9.1 Active Defense Against PDoS Attacks

Since IoCT devices are part of CPS, they also require physical "bandwidth." As an example, consider the navigation app Waze [186]. Waze uses real-time traffic information to find optimal navigation routes. Due to its large number of users, the app also influences traffic. If too many users are directed to one road, they can consume the physical bandwidth of that road and cause unexpected congestion.

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_9

Fig. 9.1 Conceptual diagram of a PDoS attack. (1) Attack sponsor hires a botnet herder. (2) Botnet herder uses the server to manage recruitment. (3) Malware scans for vulnerable IoCT devices and begins cascading infection. (4) Botnet herder uses devices (e.g., HVAC controllers) to deplete bandwidth of a cyber-physical service (e.g., electrical power)

An attacker with insider access to Waze could use this mechanism to manipulate transportation networks.

Another example can be found in healthcare. Smart lighting systems (which deploy, e.g., *time-of-flight* sensors) detect falls of room occupants [187]. These systems alert emergency responders about a medical situation in an assisted living center or the home of someone who is aging. But an attacker could potentially trigger many of these alerts at the same time, depleting the response bandwidth of emergency personnel.

Such a threat could be called a denial of *cyber-physical* service attack. To distinguish it from a cyber-layer DDoS, we also use the acronym *PDoS* (*Physical* Denial-of-Service). Figure 9.1 gives a conceptual diagram of a PDoS attack. In the rest of the chapter, we will consider one specific instance of a PDoS attack, although our analysis is not limited to this example. We consider the infection and manipulation of a population of IoCT-based heating, ventilation, and air conditioning (HVAC) controllers in order to cause a sudden load shock to the power grid. Attackers either disable demand response switches used for reducing peak load [188], or they unexpectedly activate inactive loads. This imposes risks ranging from frequency droop to load shedding and cascading failures.

9.1.1 Modeling the PDoS Recruitment Stage

Defenses against PDoS can be designed at multiple layers (Fig. 9.2). The scope of this chapter is limited to defense at the stage of botnet recruitment, in which the attacker scans a wide range of IP addresses, searching for devices with weak security settings. Mirai, for example, does this by attempting logins with a dictionary of factory-default usernames and passwords (e.g., `root/admin, admin/admin,`

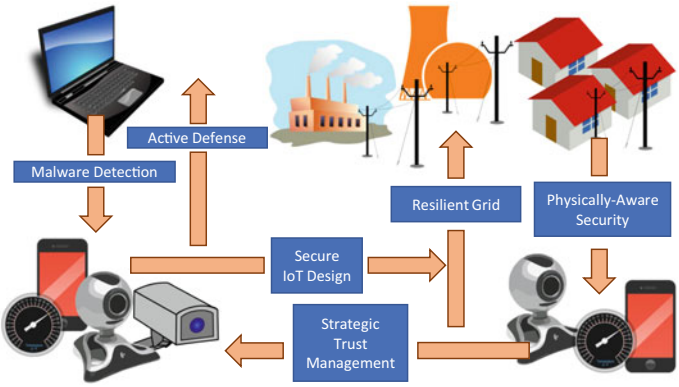


Fig. 9.2 PDoS defense can be designed at multiple layers. Malware detection and active defense can combat initial infection, secure IoCT design and strategic trust can reduce the spread of the malware, and CPS can be resilient and physically-aware. We focus on detection and active defense

`root/123456`) [189]. Devices in our mechanism identify these suspicious login attempts and use active defense to learn about the attacker or report his activity.

In order to quantify the risk of malware infection, we combine two game-theoretic models known as signaling games [38, 190] and Poisson games [191]. Signaling games model interactions between two parties, one of which possesses information unknown to the other party. While signaling games consider only two players, we extend this model by allowing the number of target IoCT devices to be a random variable (r.v.) that follows a Poisson distribution. This captures the fact that the malware scans a large number of targets. Moreover, we allow the targets to have heterogeneous abilities to detect malicious login attempts.

In Sect. 9.2, we review signaling games and Poisson games. In Sect. 9.3, we combine them to create Poisson signaling games (PSG). In Sect. 9.4, we apply PSG to quantify the population risk due to PDoS attacks. Section 9.5 obtains the PBNE of the model. Some of these equilibria are harmful to power companies and IoCT users. Therefore, we design proactive mechanisms to improve the equilibria in Sect. 9.6. We underline the key contributions in Sect. 9.7.

9.2 Signaling Games and Poisson Games

This section reviews signaling games with evidence and introduces Poisson games. In Sect. 9.3, we combine them to create PSG. PSG can be used to model many one-to-many signaling interactions in addition to PDoS. Table 9.1 summarizes the notation for this chapter.

Table 9.1 Notation for this chapter

Notation	Meaning
S and R	Sender and receiver
$\mathbf{G_S} = \left(\Theta, \mathbb{M}, \mathbb{EV}, \mathbb{A}, q^S, \delta, \tilde{u}^S, \tilde{u}^R\right)$	Signaling game tuple
$\mathbf{G_P} = \left(\lambda, \Phi, q^R, \mathbb{A}, \hat{u}^R\right)$	Poisson game tuple
$\mathbf{G_{PS}} = \left(\Theta, \Phi, \mathbb{M}, \mathbb{EV}, \mathbb{A}, \lambda, q, \delta, u^S, u^R\right)$	Complete Poisson signaling game
Θ and Φ	Set of types of S and R
$\mathbb{M}$, $\mathbb{EV}$, and $\mathbb{A}$	Sets of messages, evidence, and actions
λ	Poisson parameter
$q(\theta, \phi) = \left[\, q^S(\theta)\ q^R(\phi) \,\right]^T$	Prior probabilities of types of S and R
$\delta\left(e \mid \theta, m\right) = \left[\delta_\phi\left(e \mid \theta, m\right)\right]_{\phi \in \Phi}$	Detector probabilities
$u^S\left(m, c\right) = \left[u^S_\theta\left(m, c\right)\right]_{\theta \in \Theta}$	Utility functions of types of S
$u^R\left(\theta, m, a, c\right) = \left[u^R_\phi\left(\theta, m, a, c\right)\right]_{\phi \in \Phi}$	Utility functions of types of R
$\sigma^S_\theta(m),\ \theta \in \Theta$	Strategy of S of type x
$\sigma^R_\phi(a \mid m, e),\ \phi \in \Phi$	Strategy of R of type y
$\mu^R_\phi(\theta \mid m, e)$	Belief of R of type y
$U^S_\theta(\sigma^S_\theta, \sigma^R),\ U^R_\phi(\eta, \sigma^R \mid m, e, \mu^R_\phi)$	Mixed-strategy expected utilities
$BR^R_\phi\left(\sigma^R \mid m, e, \mu^R_\phi\right)$	Best response of R to other types
$TD^R_v\left(u^R_v, \delta_v\right),\ BP^S_d\left(\omega_d, q^R, \delta\right)$	Utility function meta-parameters

9.2.1 Signaling Games with Evidence

Signaling games are a class of dynamic, two-player, information-asymmetric games between a sender S and a receiver R (c.f. [38, 190]). Signaling games *with evidence* (introduced in Chap. 6) extend the typical definition by giving receivers some exogenous ability to detect deception[1] [135]. They are characterized by the tuple

$$\mathbf{G_S} = \left(\Theta, \mathbb{M}, \mathbb{EV}, \mathbb{A}, q^S, \delta, \tilde{u}^S, \tilde{u}^R\right).$$

First, S possess some private information unknown to R. This private information is called a *type*. The type could represent, e.g., a preference, a technological capability, or a malicious intent. Let the finite set Θ denote the set of possible types, and let $\theta \in \theta$ denote one particular type. Each type occurs with a probability $q^S(\theta)$, where $q^S : \Theta \to [0, 1]$ such that (s.t.) $\sum_{\theta \in \Theta} q^S(\theta) = 1$ and $\forall \theta \in \Theta,\ q^S(\theta) \geq 0$.

Based on his private information, S communicates a *message* to the receiver. The message could be, e.g., a pull request, the presentation of a certificate, or the

[1] This is based on the idea that deceptive senders have a harder time communicating some messages than truthful senders. In interpersonal deception, for instance, lying requires a high cognitive load, which may manifest itself in external gestures [34].

execution of an action which partly reveals the type. Let the finite set $\mathbb{M}$ denote the set of possible messages, and let $m \in \mathbb{M}$ denote one particular type. In general, S can use a strategy in which he chooses various m with different probabilities. We will introduce notation for these *mixed strategies* later.

In typical signaling games (e.g., Lewis signaling games [38, 190] and signaling games discussed by Crawford and Sobel [38]), R only knows about θ through m. But this suggests that deception is undetectable. Instead, signaling games with evidence include a *detector*[2] which emits evidence $e \in \mathbb{EV}$ about the sender's type [135]. Let $\delta : \mathbb{EV} \to [0, 1]$ s.t. for all $\theta \in \Theta$ and $m \in \mathbb{M}$, we have $\sum_{e \in \mathbb{EV}} \delta(e \mid \theta, m) = 1$ and $\delta(e \mid \theta, m) \geq 0$. Then $\delta(e \mid \theta, m)$ gives the probability with which the detector emits evidence e given type θ and message m. This probability is fixed, not a decision variable. Finally, $\mathbb{A}$ be a finite set of *actions*. Based on m and e, R chooses some $a \in \mathbb{A}$. For instance, R may choose to accept or reject a request represented by the message. These can also be chosen using a mixed strategy.

In general, θ, m, and a can impact the utility of S and R. Therefore, let $\tilde{u}^S : \mathbb{M} \times \mathbb{A} \to \mathbb{R}^{|\Theta|}$ be a vector-valued function such that $\tilde{u}^S(m, a) = \left[\tilde{u}^S_\theta(m, a)\right]_{\theta \in \Theta}$. This is a column vector with entries $\tilde{u}^S_\theta(m, a)$. These entries give the utility that S of each receiver of type $\theta \in \Theta$ obtains for sending a message m when the receiver plays action a. Next, define the utility function for R by $\tilde{u}^R : \Theta \times \mathbb{M} \times \mathbb{A} \to \mathbb{R}$, such that $\tilde{u}^R(\theta, m, a)$ gives the utility that R receives when a sender of type θ transmits message m and R plays action a.

9.2.2 Poisson Games

Poisson games were introduced by Roger Myerson in 1998 [191]. This class of games models interactions between an unknown number of players, each of which belongs to one type in a finite set of types. Modeling the population uncertainty using a Poisson r.v. is convenient because merging or splitting Poisson r.v. results in r.v. which also follow Poisson distributions.

Section 9.3 will combine signaling games with Poisson games by considering a sender which issues a command to a pool of an unknown number of receivers, which all respond at once. Therefore, let us call the players of the Poisson game "receivers," although this is not the nomenclature used in the original game. Poisson games are characterized by the tuple

$$\mathbf{G_P} = \left(\lambda, \Phi, q^R, \mathbb{A}, \hat{u}^R\right).$$

First, the population parameter $\lambda > 0$ gives the mean and variance of the Poisson distribution. For example, λ may represent the expected number of mobile phone

[2] This could literally be a hardware or software detector, such as email filters which attempt to tag phishing emails. But it could also be an abstract notion meant to signify the innate ability of a person to recognize deception.

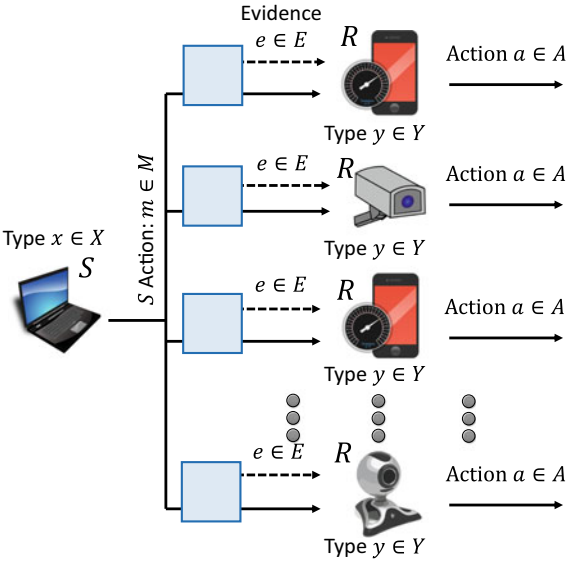


Fig. 9.3 PSG model the third stage of a PDoS attack. A sender of type θ chooses an action m which is observed by an unknown number of receivers. The receivers have multiple types $\phi \in \Phi$. Each type may observe different evidence $e \in \mathbb{EV}$. Based on m and e, each type of receiver chooses an action a

users within the range of a base station. Let the finite set Φ denote the possible types of each receiver, and let $\phi \in \Phi$ denote one of these types. Each receiver has type ϕ with probability $q^R(\phi)$, where $\sum_{\phi \in \Phi} q^R(\phi) = 1$ and $\forall \phi \in \Phi,\ q^R(\phi) > 0$.

Because of the decomposition property of the Poisson r.v., the number of receivers of each type $\phi \in \Phi$ also follows a Poisson distribution. Based on her type, each receiver chooses an action a in the finite set $\mathbb{A}$. We have deliberately used the same notation as the action for the signaling game, because these two actions will coincide in the combined model.

Utility functions in Poisson games are defined as follows. For $a \in \mathbb{A}$, let $c_a \in \mathbb{Z}_+$ (the set of nonnegative integers) denote the count of receivers which play action a. Then let c be a column vector which contains entries c_a for each $a \in \mathbb{A}$. Then c falls within the set $\mathbb{Z}(\mathbb{A})$, the set of all possible integer counts of the number of players which take each action.

Poisson games assume that all receivers of the same type receive the same utility. Therefore, let $\hat{u}^R : \mathbb{A} \times \mathbb{Z}(C) \to \mathbb{R}^{|\Phi|}$ be a vector-valued function such that $\hat{u}^R(a, c) = \left[\hat{u}^R_\phi(a, c)\right]_{\phi \in \Phi}$. The entries $\hat{u}^R_\phi(a, c)$ give the utility that receivers of each type $\phi \in \Phi$ obtain for playing an action a while the vector of the total count of receivers that play each action is given by c. Note that this is different from the utility function of receivers in the signaling game. Given the strategies of the receivers, c is also distributed according to a Poisson r.v.

9.3 Poisson Signaling Games

Figure 9.3 depicts PSG. PSG are characterized by combining $\mathbf{G_P}$ and $\mathbf{G_S}$ to obtain the tuple

$$\mathbf{G_{PS}} = \left(\Theta, \Phi, \mathbb{M}, \mathbb{EV}, \mathbb{A}, \lambda, q, \delta, u^S, u^R\right).$$

9.3.1 Types, Actions, and Evidence, and Utility

As with signaling games and Poisson games, Θ denotes the set of types of S, and Φ denotes the set of types of R. $\mathbb{M}$, $\mathbb{EV}$, and $\mathbb{A}$ denote the set of messages, evidence, and actions, respectively. The Poisson parameter is λ.

The remaining elements of $\mathbf{G_{PS}}$ are slightly modified from the signaling game or Poisson game. First, $q : \Theta \times \Phi \to [0, 1]^2$ is a vector-valued function such that $q(\theta, \phi)$ gives the probabilities $q^S(\theta)$, $\theta \in \Theta$, and $q^R(\phi)$, $\phi \in \Phi$, of each type of sender and receiver, respectively.

As in the signaling game, δ characterizes the quality of the deception detector. But receivers differ in their ability to detect deception. Various email clients, for example, may have different abilities to identify phishing attempts. Therefore, in PSG, we define the mapping by $\delta : \mathbb{EV} \to [0, 1]^{|\Phi|}$, s.t. the vector $\delta(e \mid \theta, m) = \left[\delta_\phi(e \mid \theta, m)\right]_{\phi \in \Phi}$ gives the probabilities $\delta_\phi(e \mid \theta, m)$ with which each receiver type ϕ observes evidence e given sender type θ and message m. This allows each receiver type to observe evidence with different likelihoods.[3]

The utility functions u^S and u^R are also adjusted for PSG. Let $u^S : \mathbb{M} \times \mathbb{Z}(\mathbb{A}) \to \mathbb{R}^{|\Theta|}$ be a vector-valued function s.t. the vector $u^S(m, c) = \left[u^S_\theta(m, c)\right]_{\theta \in \Theta}$ gives the utility of senders of each type θ for sending message m if the count of receivers which choose each action is given by c. Similarly, let $u^R : \Theta \times \mathbb{M} \times \mathbb{A} \times \mathbb{Z}(\mathbb{A}) \to \mathbb{R}^{|\Phi|}$ be a vector-valued function s.t. $u^R(\theta, m, a, c) = \left[u^R_\phi(\theta, m, a, c)\right]_{\phi \in \Phi}$ gives the utility of receivers of each type $\phi \in \Phi$. As earlier, θ is the type of the sender, and m is the message. But note that a denotes the action of *this particular receiver*, while c denotes the count of overall receivers which choose each action.

9.3.2 Mixed-Strategies and Expected Utility

Next, we define the nomenclature for mixed-strategies and expected utility functions. For senders of each type $\theta \in \Theta$, let $\sigma^S_\theta : \mathbb{M} \to [0, 1]$ be a mixed strategy such that $\sigma^S_\theta(m)$ gives the probability with which he plays each message $m \in \mathbb{M}$. For each

[3] In fact, although all receivers with the same type ϕ have the same likelihood $\delta_\phi(e \mid \theta, m)$ of observing evidence e given sender type θ and message m, our formulation allows the receivers to observe different actual realizations e of the evidence.

$\theta \in \Theta$, let Σ_θ^S denote the set of possible σ_θ^S. We have

$$\Sigma_\theta^S = \left\{ \bar{\sigma} \mid \sum_{m \in \mathbb{M}} \bar{\sigma}(m) = 1 \text{ and } \forall m \in \mathbb{M},\ \bar{\sigma}(m) \geq 0 \right\}.$$

For receivers of each type $\phi \in \Phi$, let $\sigma_\phi^R : \mathbb{A} \to [0, 1]$ denote a mixed strategy such that $\sigma_\phi^R(a \mid m, e)$ gives the probability with which she plays action a after observing message m and action e. For each $\phi \in \Phi$, the function σ_ϕ^R belongs to the set

$$\Sigma_\phi^R = \left\{ \bar{\sigma} \mid \sum_{a \in \mathbb{A}} \bar{\sigma}(a) = 1 \text{ and } \forall a \in \mathbb{A},\ \bar{\sigma}(a) \geq 0 \right\}.$$

In order to choose her actions, R forms a belief about the sender type θ. Let $\mu_\phi^R(\theta \mid m, e)$ denote the likelihood with which each R of type ϕ who observes message m and evidence e believes that S has type θ. In equilibrium, we will require this belief to be consistent with the strategy of S.

Now we define the expected utilities that S and each R receive for playing mixed strategies. Denote the expected utility of a sender of type $\theta \in \Theta$ by $U_\theta^S : \Sigma_\theta^S \times \Sigma^R \to \mathbb{R}$. Notice that all receiver strategies must be taken into account. This expected utility is given by

$$U_\theta^S(\sigma_\theta^S, \sigma^R) = \sum_{m \in \mathbb{M}} \sum_{c \in \mathbb{Z}(\mathbb{A})} \sigma_\theta^S(m)\, \mathbb{P}\left\{c \mid \sigma^R, \theta, m\right\} u_\theta^S(m, c).$$

Here, $\mathbb{P}\{c \mid \sigma^R, \theta, m\}$ is the probability with which the vector c gives the count of receivers that play each action. Myerson showed that, due to the aggregation and decomposition properties of the Poisson r.v., the entries of c are also Poisson r.v. [191]. Therefore, $\mathbb{P}\{c \mid \sigma^R, \theta, m\}$ is given by

$$\mathbb{P}\left\{c \mid \sigma^R, \theta, m\right\} = \prod_{a \in \mathbb{A}} e^{\lambda_a} \frac{\lambda_a^{c_a}}{c_a!},\ \lambda_a = \lambda \sum_{\phi \in \Phi} \sum_{e \in \mathbb{EV}} q^R(\phi)\, \delta_\phi(e \mid \theta, m)\, \sigma_\phi^R(a \mid m, e). \tag{9.1}$$

Next, denote the expected utility of each receiver of type $\phi \in \Phi$ by $U_\phi^R : \Sigma_\phi^R \times \Sigma^R \to \mathbb{R}$. Here, $U_\phi^R(\eta, \sigma^R \mid m, e, \mu_\phi^R)$ gives the expected utility when this particular receiver plays the mixed strategy $\eta \in \Sigma_\phi^R$ and the population of all types of receivers plays the mixed-strategy vector σ^R. The expected utility is given by

$$U_\phi^R(\eta, \sigma^R \mid m, e, \mu_y^R) = \sum_{\theta \in \Theta} \sum_{a \in \mathbb{A}} \sum_{c \in \mathbb{Z}(\mathbb{A})} \mu_\phi^R(\theta \mid m, e)\, \eta(a \mid m, e)\, \mathbb{P}\left\{c \mid \sigma^R, \theta, m\right\} u_\phi^R(\theta, m, a, c), \tag{9.2}$$

where again $\mathbb{P}\{c \mid \sigma^R, \theta, m\}$ is given by Eq. (9.1).

9.3.3 Perfect Bayesian Nash Equilibrium

First, since PSG involve sequential actions, we use an equilibrium concept which requires *perfection*. Strategies at each information set of the game must be optimal for the remaining subgame [90]. Second, since PSG involve incomplete information, we use a *Bayesian* concept. Third, since each receiver chooses her action without knowing the actions of the other receivers, the Poisson stage of the game involves a *fixed point*. All receivers choose strategies which best respond to the optimal strategies of the other receivers. PBNE is the appropriate concept for games with these criteria [90].

Consider the two chronological stages of PSG. The second stage takes place among the receivers. This stage is played with a given m, e, and μ^R determined by the sender (and detector) in the first stage of the game. When m, e, and μ^R are fixed, the interaction between all receivers becomes a standard Poisson game. Define $BR_\phi^R : \Sigma^R \to \mathcal{P}(\Sigma_\phi^R)$ (where $\mathcal{P}(\mathbb{S})$ denotes the power set of $\mathbb{S}$) such that the best response of a receiver of type ϕ to a strategy profile σ^R of the other receivers is given by the strategy or set of strategies

$$BR_\phi^R\left(\sigma^R \mid m, e, \mu_\phi^R\right) \triangleq \underset{\eta \in \Sigma_\phi^R}{\arg\max}\, U_\phi^R\left(\eta, \sigma^R \mid m, e, \mu_\phi^R\right). \tag{9.3}$$

The first stage takes place between the sender and the set of receivers. If we fix the set of receiver strategies σ^R, then the problem of a sender of type $\theta \in \Theta$ is to choose σ_θ^S to maximize his expected utility given σ^R. The last criteria is that the receiver beliefs μ^R must be consistent with the sender strategies according to Bayes' Law. Definition 9.1 applies PBNE to PSG.

Definition 9.1 (*PBNE*) Strategy and belief profile $(\sigma^{S*}, \sigma^{R*}, \mu^R)$ is a PBNE of a PSG if all of the following hold [90]:

$$\forall \theta \in \Theta,\ \sigma_\theta^{S*} \in \underset{\sigma_\theta^S \in \Sigma_\theta^S}{\arg\max}\, U_\theta^S(\sigma_\theta^S, \sigma^{R*}), \tag{9.4}$$

$$\forall \phi \in \Phi,\ \forall m \in \mathbb{M},\ \forall e \in \mathbb{EV},\ \sigma_\phi^{R*} \in BR_\phi^R\left(\sigma^{R*} \mid m, e, \mu_\phi^R\right), \tag{9.5}$$

$$\forall \phi \in \Phi,\ \forall m \in \mathbb{M},\ \forall e \in \mathbb{EV},\ \mu_\phi^R(d \mid m, e) \in \frac{\delta_\phi(e \mid d, m)\, \sigma_d^S(m)\, q^S(d)}{\sum\limits_{\tilde{\theta} \in \Theta} \delta_\phi\left(e \mid \tilde{\theta}, m\right) \sigma_{\tilde{\theta}}^S(m)\, q^S\left(\tilde{\theta}\right)}, \tag{9.6}$$

if $\sum\limits_{\tilde{\theta} \in \Theta} \delta_\phi\left(e \mid \tilde{\theta}, m\right) \sigma_{\tilde{\theta}}^S(m)\, q^S\left(\tilde{\theta}\right) > 0$, and $\mu_\phi^R(d \mid m, e) \in [0, 1]$, otherwise. We also always have $\mu_\phi^R(l \mid m, e) = 1 - \mu_\phi^R(d \mid m, e)$.

Equation (9.4) requires the sender to choose an optimal strategy given the strategies of the receivers. Based on the message and evidence that each receiver observes,

Table 9.2 Application of PSG to PDoS recruitment

Set	Elements
Type $\theta \in \Theta$ of S	l : legitimate, d : malicious
Type $\phi \in \Phi$ of R	k : no detection; o : detection; v : detection and active defense
Message $m \in \mathbb{M}$ of S	$m = \{m^1, m^2, \ldots\}$, a set of $\lvert m \rvert$ password strings
Evidence $e \in \mathbb{EV}$	b : suspicious, n : not suspicious
Action $a \in \mathbb{A}$ of R	t : trust, g : lockout, f : active defense

Eq. (9.5) requires each receiver to respond optimally to the profile of the strategies of the other receivers. Equation (9.6) uses Bayes' law (when possible) to obtain the posterior beliefs μ^R using the prior probabilities q^S, the sender strategies σ^S, and the characteristics δ_ϕ, $\phi \in \Phi$ of the detectors [135].

9.4 Application of PSG to PDoS

Section 9.3 defined PSG in general, without specifying the members of the type, message, evidence, or action sets. In this section, we apply PSG to the recruitment stage of PDoS attacks. Table 9.2 summarizes the nomenclature.

S refers to the agent which attempts a login attempt, while R refers to the device. Let the set of sender types be given by $\Theta = \{l, d\}$, where l represents a legitimate login attempt, while d represents a malicious attempt. Malicious senders use a broad IP scan to attempt to login to a large number of devices. This number is drawn from a Poisson r.v. with parameter λ. Legitimate S only attempt to login to one device at a time. Let the receiver types be $\Phi = \{k, o, v\}$. Type k represents weak receivers which have no ability to detect deception and do not use active defense. Type o represents strong receivers which can detect deception, but do not use active defense. Finally, type v represents active receivers which can both detect deception and use active defense.

9.4.1 Messages, Evidence Thresholds, and Actions

Messages consist of sets of consecutive unsuccessful login attempts. They are denoted by $m = \{m^1, m^2, \ldots\}$, where each $m^1, m^2, \ldots$ is a string entered as an attempted password.[4] For instance, botnets similar to Mirai choose a list of default passwords such as [189]

[4] A second string can also be considered for the username.

$$m = \{\texttt{admin}, \texttt{888888}, \texttt{123456}, \texttt{default}, \texttt{support}\}.$$

Of course, devices can lockout after a certain number of unsuccessful login attempts. Microsoft Server 2012 recommends choosing a threshold at 5–9 [192]. Denote the lower end of this range by $\tau_L = 5$. Let us allow all attempts with $|m| < \tau_L$. In other words, if a user successfully logs in before τ_L, then the PSG does not take place. (See Fig. 9.5.)

The PSG takes place for $|m| \geq \tau_L$. Let $\tau_H = 9$ denote the upper end of the Microsoft range. After τ_L, S may persist with up to τ_H login attempts, or he may not persist. Let p denote persist, and w denote not persist. Our goal is to force malicious S to play w with high probability.

For R of types o and v, if S persists and does not successfully log in with $|m| \leq \tau_H$ login attempts, then $e = b$. This signifies a suspicious login attempt. If S persists and does successfully login with $|m| \leq \tau_H$ attempts, then $e = n$, i.e., the attempt is not suspicious.[5]

If a user persists, then the device R must choose an action a. Let $a = t$ denote trusting the user, i.e., allowing login attempts to continue. Let $a = g$ denote locking the device to future login attempts. Finally, let $a = f$ denote using an active defense such as reporting the suspicious login attempt to an Internet service provider (ISP), recording the attempts in order to gather information about the possible attacker, or attempting to block the offending IP address.

9.4.2 Characteristics of PDoS Utility Functions

The nature of PDoS attacks implies several features of the utility functions u^S and u^R. These are listed in Table 9.3. Characteristic 1 (C1) states that if S does not persist, both players receive zero utility. C2 says that R also receives zero utility if S persists and R locks down future logins. Next, C3 states that receivers of all types receive positive utility for trusting a benign login attempt, but negative utility for trusting a malicious login attempt. We have assumed that only type v receivers use active defense; this is captured by C4. Finally, C5 says that type v receivers obtain positive utility for using active defense against a malicious login attempt, but negative utility for using active defense against a legitimate login attempt. Clearly, C1–C5 are all natural characteristics of PDoS recruitment.

[5] For strong and active receivers, $\delta_\phi(b \mid d, p) > \delta_\phi(b \mid l, p)$, $\phi \in \{o, v\}$. That is, these receivers are more likely to observe suspicious evidence if they are interacting with a malicious sender than if they are interacting with a legitimate sender. Mathematically, $\delta_k(b \mid d, p) = \delta_k(b \mid l, p)$ signifies that type k receivers do not implement a detector.

Table 9.3 Characteristics of PDoS utility functions

#	Notation
C1	$\forall\theta \in \Theta,\ \phi \in \Phi\ , a \in \mathbb{A},\ c \in \mathbb{Z}(\mathbb{A})\ ,$ $u_\theta^S(w, c) = u_\phi^R(\theta, w, a, c) = 0$
C2	$\forall\theta \in \Theta,\ \phi \in \Phi,\ c \in \mathbb{Z}(\mathbb{A})\ ,$ $u_\phi^R(\theta, p, g, c) = 0$
C3	$\forall\phi \in \Phi,\ c \in \mathbb{Z}(\mathbb{A})\ ,$ $u_\phi^R(d, p, t, c) < 0 < u_\phi^R(l, p, t, c)$
C4	$\forall\theta \in \Theta,\ c \in \mathbb{Z}(\mathbb{A})\ ,$ $u_k^R(\theta, p, f, c) = u_o^R(\theta, p, f, c) = -\infty$
C5	$\forall c \in \mathbb{Z}(\mathbb{A})\ ,$ $u_v^R(l, p, f, c) < 0 < u_v^R(d, p, f, c)$

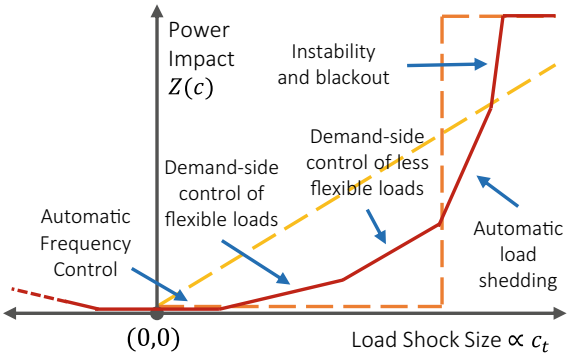


Fig. 9.4 Conceptual relationship between load shock size and damage to the power grid. Small shocks are mitigated through automatic frequency control or demand-side control of flexible loads. Large shocks can force load shedding or blackouts

9.4.3 Modeling the Physical Impact of PDoS Attacks

The quantities c_t, c_g, and c_f denote, respectively, the number of devices that trust, lockdown, and use active defense. Define the function $Z : \mathbb{Z}(\mathbb{A}) \to \mathbb{R}$ such that $Z(c)$ denotes the load shock that malicious S cause based on the count c. $Z(c)$ is clearly nondecreasing in c_t, because each device that trusts the malicious sender becomes infected and can impose some load shock to the power grid.

The red (solid) curve in Fig. 9.4 conceptually represents the mapping from load shock size to damage caused to the power grid based on the mechanisms available for regulation. Small disturbances are regulated using automatic frequency control. Larger disturbances can significantly decrease the frequency and should be mitigated. Grid operators have recently offered customers *load control switches*, which automatically deactivate appliances in response to a threshold frequency decrease [193]. But the size of this voluntary demand-side control is limited. Eventually, operators impose involuntary load shedding (i.e., rolling blackouts). This causes higher inconvenience. In the worst-case, transient instability leads to cascading failures and blackout [194].

The yellow and orange dashed curves in Fig. 9.4 provide two approximations to $Z(c)$. The yellow curve, $\tilde{Z}_{\text{lin}}(c)$, is linear in c^t. We have $\tilde{Z}_{\text{lin}}(c) = \omega_d^t c^t$, where ω_d^t is a positive real number. The orange curve, $\tilde{Z}_{\text{step}}(c)$, varies according to a step function, i.e., $Z(c) = \Omega_d^t \mathbf{1}_{\{c_t > \tau_t\}}$, where Ω_d^t is a positive real number and $\mathbf{1}_{\{\bullet\}}$ is the indicator function. In this chapter, we derive solutions for the linear approximation. Under this approximation, the utility of malicious S is given by

$$u_d^S(m, c) = \tilde{Z}_{\text{lin}}(c) + \omega_d^g c_g + \omega_d^f c_f = \omega_d^t c_t + \omega_d^g c_g + \omega_d^f c_f.$$

where $\omega_d^g < 0$ and $\omega_d^f < 0$ represent the utility to malicious S for each device that locks down or uses active defense, respectively.

Using $\tilde{Z}_{\text{lin}}(c)$, the decomposition property of the Poisson r.v. simplifies $U_\theta^S(\sigma_\theta^S, \sigma^R)$. We show in Appendix 9.9.1 that the sender's expected utility depends on the expected values of each of the Poisson r.v. that represent the number of receivers who choose each action c_a, $a \in \mathbb{A}$. The result is that

$$U_\theta^S(\sigma_\theta^S, \sigma^R) = \lambda \sigma_\theta^S(p) \sum_{\phi \in \Phi} \sum_{e \in \mathbb{EV}} \sum_{a \in \mathbb{A}} q^R(\phi)\, \delta_\phi(e \mid \theta, p)\, \sigma_\phi^R(a \mid p, e)\, \omega_\theta^a. \tag{9.7}$$

Next, assume that the utility of each receiver does not depend directly on the actions of the other receivers. (In fact, the receivers are still endogenously coupled through the action of S.) Abusing notation slightly, we drop c (the count of receiver actions) in $u_\phi^R(\theta, m, a, c)$ and σ^R (the strategies of the other receivers) in $U_\phi^R(\eta, \sigma^R \mid m, e, \mu_\phi^R)$. Equation (9.2) is now

$$U_\phi^R\left(\eta \mid m, e, \mu_\phi^R\right) = \sum_{\theta \in \Theta} \sum_{a \in \{t, f\}} \mu_\phi^R(\theta \mid m, e)\, \eta(a \mid m, e)\, u_\phi^R(\theta, m, a).$$

9.5 Equilibrium Analysis

In this section, we obtain the equilibrium results by parameter region. In order to simplify analysis, without loss of generality, let the utility functions be the same for all receiver types (except when $a = f$), i.e., $\forall \theta \in \Theta,\ u_k^R(\theta, p, t) = u_o^R(\theta, p, t) = u_v^R(\theta, p, t)$. Also without loss of generality, let the quality of the detectors for types $\phi \in \{o, v\}$ be the same: $\forall e \in \mathbb{EV},\ \theta \in \Theta,\ \delta_o(e \mid \theta, p) = \delta_v(e \mid \theta, p)$.

9.5.1 PSG Parameter Regime

We now obtain equilibria for a natural regime of the PSG parameters. First, assume that legitimate senders always persist: $\sigma_l^S(p) = 1$. This is natural for our application,

because IoCT HVAC users will always attempt to login. Second, assume that R of all types trust login attempts which appear to be legitimate (i.e., give evidence $e = n$). This is satisfied for

$$q^S(d) < \frac{u_k^R(l, p, t)}{u_k^R(l, p, t) - u_k^R(d, p, t)}. \tag{9.8}$$

Third, we consider the likely behavior of R of type o when a login attempt is suspicious. Assume that she will lockdown rather than trust the login. This occurs under the parameter regime

$$q^S(d) > \frac{\bar{u}_o^R(l, p, t)}{\bar{u}_o^R(l, p, t) - \bar{u}_o^R(d, p, t)}, \tag{9.9}$$

using the shorthand notation

$$\bar{u}_o^R(l, p, t) = u_o^R(l, p, t)\,\delta_0(b \mid l, p), \quad \bar{u}_o^R(d, p, t) = u_o^R(d, p, t)\,\delta_0(b \mid d, p).$$

The fourth assumption addresses the action of R of type v when a login attempt is suspicious. The optimal action depends on her belief $\mu_o^R(d \mid p, b)$ that S is malicious. The belief, in turn, depends on the mixed-strategy probability with which malicious S persist. We assume that there is some $\sigma_d^S(p)$ for which R should lockdown ($a = g$). This is satisfied if there exists a real number $\phi \in [0, 1]$ such that, given[6] $\sigma_d^S(p) = \phi$,

$$U_v^R(t \mid p, b, \mu_v^R) > 0, \quad U_v^R(f \mid p, b, \mu_v^R) > 0. \tag{9.10}$$

This simplifies analysis, but can be removed if necessary.

Lemma 9.2 summarizes the equilibrium results under these assumptions. Legitimate S persist, and R of type o lockdown under suspicious login attempts. All receiver types trust login attempts which appear legitimate. R of type k, since she cannot differentiate between login attempts, trusts all of them. The proof follows from the optimality conditions in Eqs. (9.4)–(9.6) and the assumptions in Eqs. (9.8)–(9.10).

Lemma 9.2 (Constant PBNE Strategies) *If $\sigma_d^S(p) = 1$ and Eqs. (9.8)–(9.10) hold, then the following equilibrium strategies are implied:*

$$\sigma_l^{S*}(p) = 1, \ \sigma_o^{R*}(g \mid p, b) = 1, \ \sigma_k^{R*}(t \mid p, b) = 1,$$

$$\sigma_o^{R*}(t \mid p, n) = \sigma_v^{R*}(t \mid p, n) = \sigma_k^{R*}(t \mid p, n) = 1.$$

Figure 9.5 depicts the results of Lemma 9.2. The remaining equilibrium strategies to be obtained are denoted by the red items for S and the blue items for R. These strategies are $\sigma_o^{R*}(\bullet \mid p, b)$, $\sigma_v^{R*}(\bullet \mid p, b)$, and $\sigma_d^{S*}(p)$. Intuitively, $\sigma_d^{S*}(p)$ depends

[6] We abuse notation slightly to write $U_v^R(a \mid m, e, \mu_\phi^R)$ for the expected utility that R of type v obtains by playing action a.

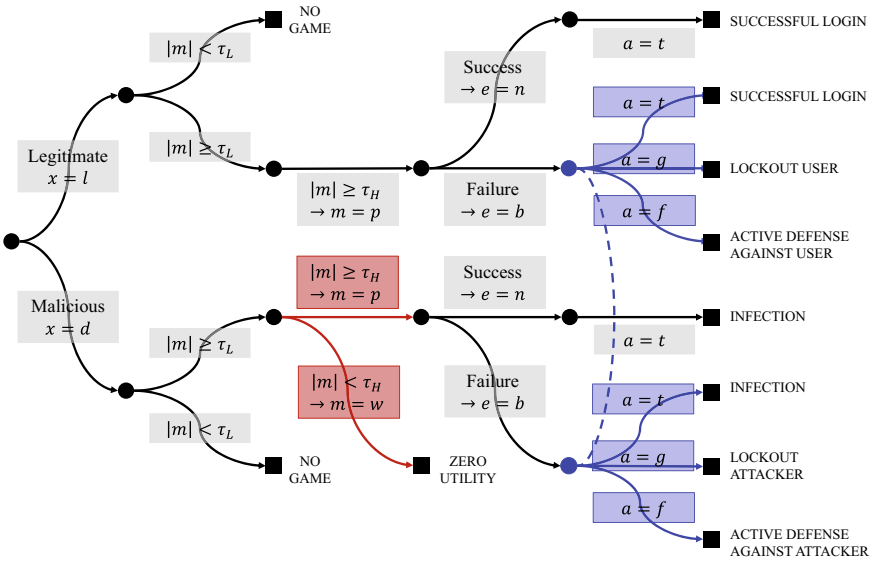


Fig. 9.5 Model of a PSG under Lemma 9.2. Only one of many R is depicted. After the types θ and ϕ, of S and R, respectively, are drawn, S chooses whether to persist beyond τ_L attempts. Then R chooses to trust, lockout, or use active defense against S based on whether S is successful. Lemma 9.2 determines all equilibrium strategies except $\sigma_d^{S*}(\bullet)$, $\sigma_o^{R*}(\bullet \mid p, b)$, and $\sigma_v^{R*}(\bullet \mid p, b)$, marked by the blue and red items

on whether R of type o and type v will lockdown and/or use active defense to oppose suspicious login attempts.

9.5.2 *Equilibrium Strategies*

The remaining equilibrium strategies fall into four parameter regions. In order to delineate these regions, we define two quantities.

Let $TD_v^R(u_v^R, \delta_v)$ denote a threshold which determines the optimal action of R of type v if $\sigma_d^S(p) = 1$. If $q^S(d) > TD_v^R(u_v^R, \delta_v)$, then the receiver uses active defense with some probability. Equation (9.3) can be used to show that

$$TD_v^R\left(u_v^R, \delta_v\right) = \frac{\bar{u}_v^R(l, p, f)}{\bar{u}_v^R(l, p, f) - \bar{u}_v^R(d, p, f)},$$

where we have used the shorthand notation

$$\bar{u}_v^R(l, p, f) := u_v^R(l, p, f)\,\delta_v(b \mid l, p), \quad \bar{u}_v^R(d, p, f) := u_v^R(d, p, f)\,\delta_v(b \mid d, p).$$

Table 9.4 Equilibrium regions of the PSG for PDoS

	$q^S(d) < TD_v^R(\bullet)$	$q^S(d) > TD_v^R(\bullet)$
$BP_d^S(\bullet \mid t, g, g) < 0$,	Vulnerable attacker	
	$\sigma^{S*}(p) < 1$	
$BP_d^S(\bullet \mid t, g, f) < 0$	$0 < \sigma_o^{R*}(t \mid p, b),\ \sigma_o^{R*}(g \mid p, b) < 1$	
	$0 < \sigma_v^{R*}(t \mid p, b),\ \sigma_v^{R*}(g \mid p, b) < 1$	
$BP_d^S(\bullet \mid t, g, g) > 0$,		Active deterrence
		$\sigma^{S*}(p) < 1$
$BP_d^S(\bullet \mid t, g, f) < 0$	Status quo	$\sigma_o^{R*}(g \mid p, b) = 1$
	$\sigma^{S*}(p) = 1$	$0 < \sigma_v^{R*}(g \mid p, b)$,
	$\sigma_o^{R*}(g \mid p, b) = 1$	$\sigma_v^{R*}(f \mid p, b) < 1$
$BP_d^S(\bullet \mid t, g, g) > 0$,	$\sigma_v^{R*}(g \mid p, b) = 1$	Resistant attacker
		$\sigma^{S*}(p) = 1$
$BP_d^S(\bullet \mid t, g, f) > 0$		$\sigma_o^{R*}(g \mid p, b) = 1$
		$\sigma_v^{R*}(f \mid p, b) = 1$

Next, let $BP_d^S\left(\omega_d, q^R, \delta\right)$ denote the benefit which S of type d receives for choosing $m = p$, i.e., for persisting. We have

$$BP_d^S\left(\omega_d, q^R, \delta\right) := \sum_{\phi \in \Phi} \sum_{e \in \mathbb{EV}} \sum_{a \in \mathbb{A}} q^R(\phi)\, \delta_\phi(e \mid d, p)\, \sigma_\phi^R(a \mid p, e)\, \omega_d^a.$$

If this benefit is negative, then S will not persist. Let $BP_d^S\left(\omega_d, q^R, \delta \mid a_k, a_o, a_v\right)$ denote the benefit of persisting when receivers use the pure strategies

$$\sigma_k^R(a_k \mid p, b) = \sigma_o^R(a_o \mid p, b) = \sigma_v^R(a_v \mid p, b) = 1.$$

We now have Theorem 9.3, which predicts the risk of malware infection in the remaining parameter regions. The proof is in Appendix 9.9.2.

Theorem 9.3 (PBNE within Regions) *If $\sigma_d^S(p) = 1$ and Eqs. (9.8)–(9.10) hold, then $\sigma_o^{R*}(\bullet \mid p, b)$, $\sigma_v^{R*}(\bullet \mid p, b)$, and $\sigma_d^{S*}(p)$ vary within the four regions listed in Table 9.4.*

In the *status quo* equilibrium, strong and active receivers lockdown under suspicious login attempts. But this is not enough to deter malicious senders from persisting. We call this the status quo because it represents current scenarios in which botnets infect vulnerable devices but incur little damage from being locked out of secure devices. This is a poor equilibrium, because $\sigma_d^{S*}(p) = 1$.

In the *active deterrence* equilibrium, lockouts are not sufficient to deter malicious S from fully persisting. But since $q^S(d) > TD_v^R$, R of type v use active defense. This is enough to deter malicious S : $\sigma_d^{S*}(p) < 1$. In this equilibrium, R of type o always

locks down: $\sigma_o^{R*}(g \mid p, b) = 1$. R of type v uses active defense with probability

$$\sigma_v^{R*}(f \mid p, b) = \frac{\omega_d^t q^R(k) + \omega_d^g \left(q^R(o) + q^R(v)\right)}{\left(\omega_d^g - \omega_d^f\right) q^R(v)\, \delta_v(v \mid d, p)}, \tag{9.11}$$

and otherwise locks down: $\sigma_v^{R*}(g \mid p, b) = 1 - \sigma_v^{R*}(f \mid p, b)$. Deceptive S persist with reduced probability

$$\sigma_d^{S*}(p) = \frac{1}{q^S(d)} \left(\frac{\bar{u}_v^R(l, p, f)}{\bar{u}_v^R(l, p, f) - \bar{u}_v^R(d, p, f)} \right). \tag{9.12}$$

In the *resistant attacker* equilibrium, $q^S(d) > TD_v^R$. Therefore, R of type v use active defense. But $BP_d^S(\bullet \mid t, g, f) > 0$, which means that the active defense is not enough to deter malicious senders. This is a "hopeless" situation for defenders, since all available means are not able to deter malicious senders. We still have $\sigma_d^{S*}(p) = 1$.

In the *vulnerable attacker* equilibrium, there is no active defense. But R of type o and type v lockdown under suspicious login attempts, and this is enough to deter malicious S, because $BP_d^S(\bullet \mid t, g, g) < 0$. R of types o and v lockdown with probability

$$\sigma_o^{R*}(g \mid p, b) = \sigma_v^{R*}(g \mid p, b) = \frac{\omega_d^t}{\left(q^R(0) + q^R(v)\right) \delta_o(b \mid d, p) \left(\omega_d^t - \omega_d^g\right)}, \tag{9.13}$$

and trust with probability $\sigma_o^{R*}(t \mid p, b) = \sigma_v^{R*}(t \mid p, b) = 1 - \sigma_o^{R*}(g \mid p, b)$. Deceptive S persist with reduced probability

$$\sigma_d^{S*}(p) = \frac{1}{q^S(d)} \left(\frac{\bar{u}_o^R(l, p, t)}{\bar{u}_o^R(l, p, t) - \bar{u}_o^R(d, p, t)} \right). \tag{9.14}$$

The *status quo* and *resistant attacker* equilibria are poor results because infection of devices is not deterred at all. The focus of Sect. 9.6 will be to shift the PBNE to the other equilibrium regions, in which infection of devices is deterred to some degree.

9.6 Mechanism Design

The equilibrium results are delineated by q^S, $TD_v^R(u_v^R, \delta_v)$ and $BP_d^S(\omega_d, q^R, \delta)$. These quantities are functions of the parameters q^S, q^R, δ_o, δ_v, ω_d, and u_v^R. Mechanism design manipulates these parameters in order to obtain a desired equilibrium. We discuss two possible mechanisms.

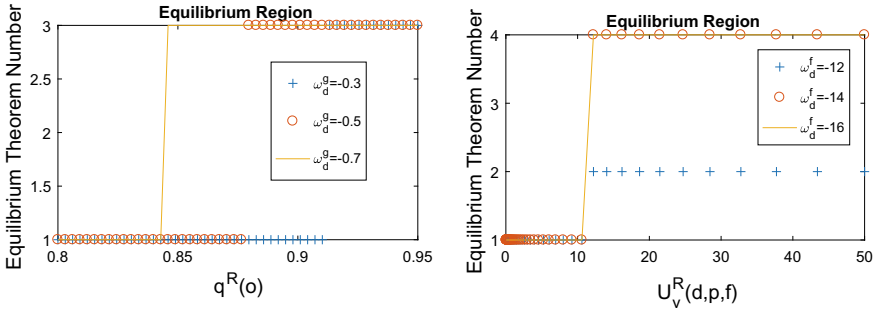


Fig. 9.6 Equilibrium transitions for **a** legal and **b** active defense mechanisms. The equilibrium numbers signify: 1-*status quo*, 2-*resistant attacker*, 3-*vulnerable attacker*, 4-*active deterrence*

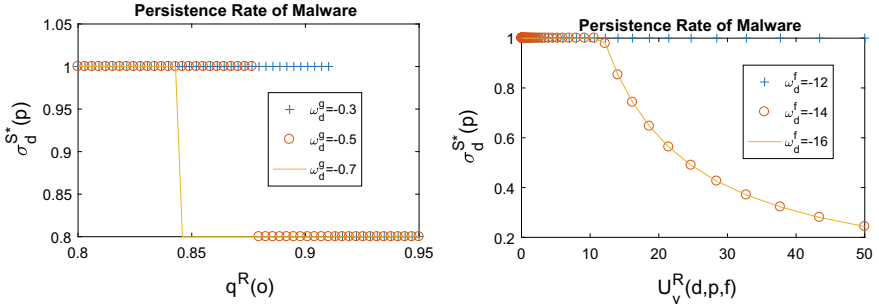


Fig. 9.7 Malware persistence rate for **a** legal and **b** active defense mechanisms

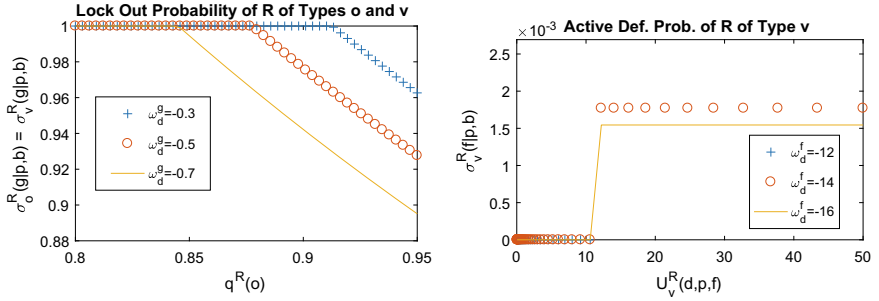


Fig. 9.8 Probabilities of opposing malicious S. Plot **a**: probability that R lockdown with the legal mechanism. Plot **b**: probability that R use active defense

9.6.1 Legislating Basic Security

Malware which infects IoCT devices is successful because many IoCT devices are poorly secured. Therefore, one mechanism design idea is to legally require better authentication methods, in order to decrease $q^R(k)$ and increase $q^R(o)$.

The left-hand sides of Figs. 9.6, 9.7 and 9.8 depict the results. Figure 9.6a shows that decreasing $q^R(k)$ and increasing $q^R(o)$ moves the game from the *status quo* equilibrium to the *vulnerable attacker* equilibrium. But Fig. 9.7a shows that this only causes a fixed decrease in $\sigma_d^{S*}(p)$, regardless of the amount of decrease in $q^R(k)$. The reason, as shown in Fig. 9.8a, is that as $q^R(o)$ increases, it is incentive-compatible for receivers to lockdown with progressively lower probability $\sigma_\phi^{R*}(g \mid p, b)$, $\phi \in \{o, v\}$. Rather than forcing malicious S to not persist, increasing $q^R(o)$ only decreases the incentive for receivers to lockdown under suspicious login attempts.

9.6.2 *Incentivizing Active Defense*

One reason for the proliferation of IoCT malware is that most devices which are secure (i.e., R of type $\phi = o$) do not take any actions against malicious login attempts except to lockdown (i.e., to play $a = g$). But there is almost no cost to malware scanners for making a large number of login attempts under which devices simply lockdown. There is a lack of economic pressure which would force $\sigma_d^{S*}(p) < 1$, unless $q^R(0) \approx 1$.

This is the motivation for using active defense such as reporting the activity to an ISP or recording the attempts in order to gather information about the attacker. The right hand sides of Figs. 9.6, 9.7 and 9.8 show the effects of providing an incentive $u_v^R(d, p, f)$ for active defense. This incentive moves the game from the *status quo* equilibrium to either the *resistant attacker* equilibrium or the *vulnerable attacker* equilibrium, depending on whether $BP_d^S(\bullet \mid t, g, f)$ is positive (Fig. 9.6b). In the *vulnerable attacker* equilibrium, the persistence rate of malicious S is decreased (Fig. 9.7b). Finally, Fig. 9.8b shows that only a small amount of active defense $\sigma_v^{R*}(f \mid p, b)$ is necessary, particularly for high values of[7] ω_d^f.

9.7 Discussion of Results

The first result is that *the defender can bound the activity level of the botnet*. Recall that the *vulnerable attacker* and *active deterrence* equilibria force $\sigma_d^{S*}(p) < 1$. That is, they decrease the persistence rate of the malware scanner. But another interpretation is possible. In Eqs. (9.14) and (9.12), the product $\sigma_d^{S*}(p)\, q^S(d)$ is bounded. This product can be understood as the *total activity* of botnet scanners: a combination of prior probability of malicious senders and the effort that malicious senders exert.[8] Bensoussan et al. note that the operators of the Conficker botnet of 2008–2009 were

[7] In Fig. 9.8b, $\sigma_v^{R*}(f \mid p, b) = 1$ for $\omega_d^f = -12$.

[8] A natural interpretation in an evolutionary game framework would be that $\sigma_d^{S*}(p) = 1$, and $q^S(d)$ decreases when the total activity is bounded. In other words, malicious senders continue recruiting, but some malicious senders drop out since not all of them are supported in equilibrium.

forced to limit its activity [195, 196]. High activity levels would have attracted too much attention. The authors of [195] confirm this result analytically, using a dynamic game based on an SIS infection model. Interestingly, our result agrees with [195], but using a different framework.

Secondly, we *compare the effects of legal and economic mechanisms to deter recruitment for PDoS*. Figures 9.6a, 9.7 and 9.8a showed that $\sigma_d^{S*}(p)$ can only be reduced by a fixed factor by mandating security for more and more devices. In this example, we found that strategic behavior worked against legal requirements. By comparison, Figs. 9.6b, 9.7 and 9.8b showed that $\sigma_d^{S*}(p)$ can be driven arbitrarily low by providing an economic incentive $u_v^R(d, p, f)$ to use active defense.

Three areas are especially attractive for future work. A dynamic model can be used to model the effect of voltage load on the power grid. We can also consider the effect of load near thresholds at which the system is highly nonlinear. Finally, we can design optimal detectors δ_o and δ_v by examining the way that equilibria change as the detector properties are varied along an ROC curve.

9.8 Related Work

Signaling games are often used to model deception and trust in cybersecurity [136, 197]. Poisson games have also been used to model malware epidemics in large populations [198]. Wu et al. use game theory to design defense mechanisms against DDoS attacks [199]. Bensoussan et al. use a susceptible-infected-susceptible (SIS) model to study the growth of a botnet [195]. Finally, *load altering attacks* [200, 201] to the power grid are an example of PDoS attacks.

9.9 Derivations

9.9.1 Simplified Sender Expected Utility

Each component of c is distributed according to a Poisson r.v. The components are independent, so $\mathbb{P}\{c \mid \sigma^R, \theta, m\} = \prod_{a \in \mathbb{A}} \mathbb{P}\{c_a \mid \sigma^R, \theta, m\}$. Recall that S receives zero utility when he plays $m = w$. So we can choose $m = p$:

$$U_\theta^S(\sigma_\theta^S, \sigma^R) = \sigma_\theta^S(p) \sum_{c \in \mathbb{Z}(\mathbb{A})} \prod_{a \in \mathbb{A}} \mathbb{P}\left\{c_a \mid \sigma^R, \theta, p\right\} \left(\omega_\theta^t c_t + \omega_\theta^g c_g + \omega_\theta^f c_f\right).$$

Some of the probability terms can be summed over their support. We are left with

$$U_\theta^S(\sigma_\theta^S, \sigma^R) = \sigma_\theta^S(p) \sum_{a \in \mathbb{A}} \omega_\theta^a \sum_{c_a \in \mathbb{Z}_+} c_a \mathbb{P}\left\{c_a \mid \sigma^R, \theta, p\right\}. \tag{9.15}$$

The last summation is the expected value of c_a, which is λ_a. This yields Eq. (9.7).

9.9.2 *Proof of Theorem 9.3*

The proofs for the *status quo* and *resistant attacker* equilibria are similar to the proof for Lemma 9.2. The *vulnerable attacker* equilibrium is a partially-separating PBNE. Strategies $\sigma_o^{R*}(g \mid p, b)$ and $\sigma_v^{R*}(g \mid p, b)$ which satisfy Eq. (9.13) make malicious senders exactly indifferent between $m = p$ and $m = w$. Thus, they can play the mixed strategy in Eq. (9.14), which makes strong and active receivers exactly indifferent between $a = g$ and $a = t$. The proof of the *vulnerable attacker* equilibrium follows a similar logic.

9.10 Notes

The model of PSG is not restricted to PDoS attacks. PSG applies to any scenario in which one sender communicates a possibly malicious or misleading message to an unknown number of receivers. In the IoCT, the model could capture the communication of a roadside location-based service to a set of autonomous vehicles, or spoofing of a GPS signal used by multiple ships with automatic navigation control, for example. Online, the model could apply to deceptive opinion spam in product reviews. In other contexts, PSG could be used to model product advertising or political messaging.

Part IV
Challenges and Opportunities in Cyber Deception

Chapter 10
Insights and Future Directions

In this chapter, we attempt to go beyond a restatement of the content of the book. First, we consider the broader impacts of this research and the lessons learned during its completion. These insights are situated within the current state of cybersecurity research, and more importantly, within current societal challenges. Second, we outline an ambitious range of future research which could be carried out using this book as a foundation.

10.1 Broader Insights

This book has attempted to make contributions in several broad categories: theoretical depth, current applications, and cross-disciplinary research. Since defensive deception is a nascent field, we have focused especially on overcoming the difficulties of laying foundations and suggesting methods for approaching problems.

One challenge of interdisciplinary and emerging research is the lack of a consensus around terms and definitions. Our taxonomy of defensive deception (Chap. 4) has suggested a set of common concepts with clear definitions. By categorizing existing work into this taxonomy, we have contributed towards an understanding of the current state of game-theoretic research in defensive deception.

Another fundamental challenge of deception research is its interdisciplinary nature. In particular, deception touches upon ethical questions and moral issues. Does obfuscation (e.g., through automating misleading search engine queries) unduly violate terms of use and waste resources? Can it convince behavioral trackers to offer privacy protection, or will it only pollute a shared pool of data that ought to be a common resource? As part of the obfuscation research in Chap. 5, we prepared a presentation for the *2017 International Workshop on Obfuscation: Science, Technology, and Theory* [202]. In this presentation, we offered a first attempt to bridge the gap

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_10

between the quantitative approach of game theory and the philosophical approach of ethics.

In any emerging field, it is important to develop a set of available models and tools. Our work on signaling games with evidence (Chap. 6) focused almost exclusively on creating a concise, minimal model. Previous research had used signaling games to model deception. But we noticed that most deception leaks some evidence to the agent being deceived, and we modeled this leakage using insights from hypothesis testing to augment the traditional signaling game. We carefully defined a model of so-called leaky deception and conducted a detailed analysis of its comparative statics. This model can serve as a foundation for applications in social engineering, online marketplaces, and fake news. As part of this chapter, we noted the feature described in Remark 10.1.

Remark 10.1 (*Aggressive detectors induce truth-telling conventions*) Aggressive detectors—those that prioritize high true-positive rates over low false-positive rates—lead to equilibrium behavior in which deceptive senders communicate honestly. The senders hope that their honest communication will trigger a false alarm and the receiver will mistakenly judge the communication to be a lie. On the other hand, conservative detectors encourage falsification, because the detectors often allow it to pass without an alarm.

Our work on strategic trust investigated so-called games-of-games frameworks. We proposed a method for an autonomous vehicle to decide whether to trust remote sources of navigation information such as other vehicles, GPS signals, and location-based services. Our framework predicted two possible long-run outcomes. In the first outcome, the vehicle would set a risk threshold and would reject all commands from the cloud that fell outside of this risk threshold. This would allow attackers to cause minor damage by sending commands within the threshold. The second outcome arose from a powerful attacker that could cause significant damage even using low-risk commands. In that case, our results predicted a limit-cycle behavior that we call the *telemarketer cycle*.

Remark 10.2 The *telemarketer cycle* uses the phenomenon of telemarketer calls to explain long-run behavior in the cloud-enabled IoCT. Imagine that telemarketers and real users make phone calls to landlines with some frequency. If the frequency of telemarketer calls becomes sufficiently high, then some people with landlines may begin to ignore incoming calls—or at least to ignore those incoming calls which do not come from a recognized number. But if this happens, then being a telemarketer becomes an unprofitable business. In that case, telemarketers drop out of the market, and the frequency of telemarketer calls decreases. Then it once more becomes incentive-compatible for landline users to answer incoming calls. This draws telemarketers back into business and the cycle continues. In this illustration, telemarketers represent cloud attackers and regular callers represent cloud defenders. As one or the other dominates, landline users (autonomous vehicles) decide whether to trust incoming calls (navigation signals from remote sources).

Chapter 9 analyzed crowd deception that is not only information-asymmetric and dynamic, but also involves an unknown number of receivers (agents being deceived)

with heterogeneous abilities to detect deception. These features motivated us to extend our model of signaling games with evidence. We considered a number of receivers that was drawn from a Poisson distribution, and a breakdown of these receivers into types with various abilities to detect deception. Our equilibrium concept for this model required the receivers to play strategies that mutually best responded to the strategies of the other receivers. They were also required to be optimal against the signaling strategy of the sender. Finally, our equilibrium concept required beliefs to be updated consistently. Chapter 9 suggested a general phenomenon that we might call the *warning label phenomenon*.

Remark 10.3 The *warning label phenomenon* states that devices do not benefit from being endowed with deception detection if there are insufficient incentives to heed warnings from the detector. As an analogy: consumer protection laws require companies to include warnings about possible dangers of their products. But some of these dangers may be so rare that it is more efficient for consumers to ignore the warnings altogether than to change their behaviors based on the warnings. Although we observed this phenomenon in the context of botnet recruitment, consumer warning labels serve as a familiar illustration.

10.2 Future Directions

One promising outcome of the research in this book is that it has discovered opportunities for an equal volume of future work. First, our taxonomy of defensive deception (Chap. 4) led us to realize the need for a similar taxonomy of counter-deception. Counter-deception could include areas such as adversarial machine learning, strategic trust, and the design of detection mechanisms. But a systematic treatment of counter-deception would clarify the relationships between these areas.

Chapter 5 included a theoretical result that users who obfuscate their data would prefer either complete obfuscation or no obfuscation at all. During our research, we spoke with a developer of *TrackMeNot* [37] (a browser extension for obfuscation) who said that some users had asked for a setting to obfuscate at an intermediate level. This points towards the need for iterative research that uses implementation results to improve upon theory and theory to improve upon implementation results.

Chapter 6 can be extended by considering a model of leaky deception with continuous message and action spaces. We can build upon existing work with continuous spaces in [38, 39, 203]. Continuous spaces are especially appropriate for cyber-physical messages such as control and measurement signals. It will be interesting to see whether results for the continuous model of leaky deception follow similar structures to those in [38, 39, 203].

The attacker engagement study in Chap. 7 focused on questions of timing by abstracting away from network structure and specific technologies used for deception. Nevertheless, future work can consider both of these elements. Network structure is important in order to design an attacker engagement policy that protects critical assets

(e.g., process control networks and databases). In addition, our model of signaling games with evidence can be incorporated in order to model the degree to which the defender is able to successfully deceive the attacker. Finally, we can incorporate control theory if the attacker's objective is to destabilize a dynamic system regulated by a process control network.

Chapter 8 studied strategic trust using a bi-level framework in which a set of `FlipIt` games and a signaling game were coupled through best-response dynamics. This was a *parallel* combination of games, in which we required the mappings that characterized each game to be satisfied simultaneously. By contrast, a *series* combination of games—in which a game begins after the previous game ends—would require mappings to be satisfied by backward induction. One area for future work is to develop a general theory of multi-game compositions. Many new interactions can be synthesized through combinations of parallel and series interconnections. Importantly, each component game can be summarized by a mapping that ignores irrelevant information about the game, and focuses on what is important for the other games and also holds robustly. In this way, designers can work with combinations of many simple mappings rather than with all of the low-resolution information about all the games.

In addition, future work can apply the physically-aware strategic trust in Chap. 8 to a specific dynamical system. While this chapter aims for maximum generality in order to provide transferable results, implementation for a specific system will produce new insights and ideas for modifications. A broad array of engineering systems are candidates for applications, including autonomous air, land, and sea vehicles, wearable biomedical devices, manufacturing facilities, 3D printers [204], power generation plants, communication-based train control [68], and others.

In Chap. 9, we considered one-to-many deception in which the agents being deceived have heterogeneous abilities to detect deception. We assumed that these agents are aware of their limited detection capability; i.e., they have *known-unknowns*. But it is also possible that some agents are unaware of their poor detection capabilities. These agents have *unknown-unknowns*. Future work can consider how to model the deception of agents with unknown-unknowns.

Chapter 11
Current Challenges in Cyber Deception

While the previous chapter focused on insights and promising directions based upon the current book, this chapter examines challenges in cyber deception that are found more broadly in the existing literature. As a point of reference, we refer to the research works mentioned in Chap. 4. The literature discussed in Chap. 4 points towards four major challenges in cyber deception.

11.1 Open Problems in Existing Literature

11.1.1 Mimesis

In the papers listed in Table 4.1 of Chap. 4, cryptic deception predominates over mimetic deception.[1] One explanation is that crypsis is the goal of all privacy research. Another explanation is that the straight-forward technique of randomization falls within the category of crypsis. Finally, research in mimesis may face greater ethical concerns than in crypsis. For instance, ought a Chief Information Officer to publish a document which falsely states the number of data servers that the company runs in order mislead an adversary? Clearly, there are some aspects of deception that could infringe on other values such as trustworthiness. These caveats notwithstanding, more opportunity remains in mimetic deception.

[1] For a complete review of the papers listed in Table 4.1, see the literature review presented in [101].

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9_11

11.1.2 Theoretical Advances

The vast majority of the papers listed in Table 4.1 use Stackelberg games and Nash games. Many combine these with rich nonstrategic models. For example, [92] combines a Stackelberg game with a Markov decision process, and [49] combines a Stackelberg game with dynamic systems analysis. Few papers, though, include advanced game-theoretic models. For instance, the literature that we surveyed did not include any cooperative games. In addition, dynamic games were not often considered, partly because general stochastic games are fundamentally difficult to analyze. This may also reflect the relative newness of research in defensive deception for cybersecurity and privacy, as well as the application of game theory to this area.

11.1.3 Practical Implementations

Game theory has been successfully deployed for physical security in several applications. For instance, the ARMOR system is a game-theoretic protocol that has been deployed at Los Angeles International Airport [78]. This protocol uses Stackelberg games. Similarly, Stackelberg games have been used in order to optimize the assignment of Federal Air Marshals to U.S. commercial flights [205]. Yet it is difficult to identify successful implementations of game-theoretic concepts to cybersecurity. Of course, it is possible that commercial endeavors use game-theoretic defenses, but prefer not to publish the results. Still, several challenges frustrate the deployment of game theory in cybersecurity.

One obstacle is that some cybersecurity professionals may be wary of so-called *security through obscurity*, i.e., deceptive security mechanisms that rely only on an attacker's lack of information. It is true that deceptive mechanisms should be combined with traditional approaches such as cryptography and access control. In our opinion, though, game theory provides precisely the right set of tools to offer security professionals *provable guarantees* on what can be achieved by deception. Many game theory models assume that the adversary has full access to the strategy of the defender. Hence, they obtain worst-case guarantees, and are not undermined if an attacker learns the defender's strategy. Indeed, some government organizations have enthusiastically adopted deception for cybersecurity. For example, American government researchers recently conducted a series of observed red-team exercises involving deception. Data collectors monitored the actions and rationale of the attackers and defenders [206]. Such exercises are invaluable in making sure that game-theoretic models accurately capture elements of human psychology.

A second obstacle is, paradoxically, the high demand for technical network security professionals. Since industry and government organizations have a high demand for security analysts, and since these analysts need to constantly monitor network traffic, it is challenging to establish collaborations between the analysts and academic researchers with backgrounds in game theory. This collaboration is crucial in order

to ensure that game theorists solve relevant problems. On the other hand, work in game theory also offers the possibility to relieve the burden on analysts. For example, game theory can be used to optimally design IDS alerts in order to limit the number of false positives that must be analyzed.

11.1.4 Interdisciplinary Security

Our paper has analyzed deception as a quantitative science. Deception, however, is interdisciplinary. Economics research offers well-developed game-theoretic models. Often, however, the challenge is to apply these models in accurate, domain-specific ways. Work in experimental economics emphasizes behavioral or sub-rational aspects of deception. These aspects are critical for applications in which the attacker does not play the theoretically-optimal strategy. Additionally, psychology can be employed to analyze attacker preferences and develop accurate threat models. Finally, criminology can be useful to detect signs of deception by humans.

11.2 Closing Remarks

Deception in cybersecurity is fundamentally difficult to study. Asymmetric information, belief consistency, and sub-optimal strategies are all significant theoretical challenges. Interdisciplinary work is needed to develop accurate models of the psychology of attackers and system operators. From a practical point of view, there is a need for successfully implemented test cases in order to provide benchmarks and increase confidence in game-theoretic approaches. The IoT is rapidly evolving, and models must be updated in order to account for new paradigms and attack strategies.

These challenges have demanded that initial research in deception for cybersecurity of the IoT focus on immediate solutions for specific systems. But as defenders reclaim lost ground, we are afforded the opportunity to study deception and cybersecurity more broadly, from a systematic point of view. In this book, we have tried to abstract from the IoT its most essential elements: dynamics, heterogeneity, modularity, decentralized control, and competition. We have used systems sciences such as game theory to capture these essential elements within quantitative models. Then we have used these models to design optimal, robust, and multi-layered approaches to defensive deception and counter-deception. This process has demanded theoretical innovations such as a game-theoretic taxonomy, mean-field Stackelberg games, signaling games with evidence, Poisson signaling games, and Gestalt Nash equilibrium. It also has helped address current challenges in dynamic honeynets, information privacy, crowd deception, and autonomous vehicles. Above all, we hope that this book has laid foundations for abundant future research in game-theoretic cyber deception.

References

1. 14 grand challenges for engineering in the 21st century. National Academy of Engineering, http://www.engineeringchallenges.org/challenges.aspx. Accessed 2 Jan 2020
2. B. Edwards, S. Hofmeyr, S. Forrest, Hype and heavy tails: a closer look at data breaches. J. Cybersecur. **2**(1), 3–14 (2016)
3. J. Pawlick, A systems science perspective on deception for cybersecurity in the Internet of Things. Ph.D. thesis, New York University (2018)
4. Internet of Things: privacy and security in a connected world. Technical report, Federal Trade Commission (2015)
5. S.R. Peppet, Regulating the Internet of Things: first steps toward managing discrimination, privacy, security, and consent. Tex. Law Rev. **93**, 85 (2014)
6. Visions and challenges for realising the Internet of Things. Technical report, CERP-IoT Cluster, European Commission (2010)
7. J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An information framework for creating a smart city through Internet of Things. IEEE Internet of Things J. **1**(2), 112–121 (2014)
8. Auto-ID Labs, http://autoidlabs.org/
9. L. Atzori, A. Iera, G. Morabito, The Internet of Things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
10. H. Vogt, Efficient object identification with passive RFID tags, in *International Conference on Pervasive Computing* (Springer, 2002), pp. 98–113
11. H. Ma, D. Zhao, P. Yuan, Opportunities in mobile crowd sensing. IEEE Commun. Mag. **52**(8), 29–35 (2014)
12. M.O. Jackson, A. Wolinsky, A strategic model of social and economic networks. J. Econ. Theory **71**(1), 44–74 (1996)
13. Q. Zhang, M.F. Zhani, S. Zhang, Q. Zhu, R. Boutaba, J.L. Hellerstein, Dynamic energy-aware capacity provisioning for cloud computing environments, in *Proceedings of the 9th International Conference on Autonomic Computing* (2012), pp. 145–154
14. Q. Zhang, Q. Zhu, M.F. Zhani, R. Boutaba, J.L. Hellerstein, Dynamic service placement in geographically distributed clouds. IEEE J. Sel. Areas Commun. **31**(12), 762–772 (2013)
15. F. Bonomi, R. Milito, J. Zhu, S. Addepalli, Fog computing and its role in the Internet of Things, in *Proceedings of the MCC Workshop on Mobile Cloud Computing* (ACM, 2012), pp. 13–16
16. M. Swan, Sensor Mania! the Internet of Things, wearable computing, objective metrics, and the quantified self 2.0. J. Sens. Actuator Netw. **1**, 217–253 (2012)

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9

17. J. Meyerowitz, R. Roy Choudhury, Hiding stars with fireworks: location privacy through camouflage, in *Proceedings of the International Conference on Mobile Computing and Networking* (ACM, 2009), pp. 345–356
18. J. Pawlick, Q. Zhu, A Stackelberg game perspective on the conflict between machine learning and data obfuscation, in *IEEE Workshop on Information Forensics and Security* (2016), pp. 1–6
19. A. Nayak, I. Stojmenovic, *Wireless Sensor and Actuator Networks* (Wiley, New York, 2010)
20. J. Chen, Q. Zhu, Optimal contract design under asymmetric information for cloud-enabled internet of controlled things, *Decision and Game Theory for Security* (Springer, Berlin, 2016), pp. 329–348
21. J. Pawlick, Q. Zhu, Strategic trust in cloud-enabled cyber-physical systems with an application to glucose control. IEEE Trans. Inf. Forensics Secur. **12**(12), 2906–2919 (2017)
22. R. Baheti, H. Gill, Cyber-physical systems. Impact Control Technol. **12**, 161–166 (2011)
23. D. Kuipers, M. Fabro, Control systems cyber security: defense in depth strategies. Technical report, Idaho National Laboratory (INL) (2006)
24. S. Rass, Q. Zhu, GADAPT: a sequential game-theoretic framework for designing defense-in-depth strategies against advanced persistent threats, in *International Conference on Decision and Game Theory for Security* (Springer, 2016), pp. 314–326
25. M.H. Manshaei, Q. Zhu, T. Alpcan, T. Bacşar, J.-P. Hubaux, Game theory meets network security and privacy. ACM Comput. Surv. (CSUR) **45**(3), 25 (2013)
26. Y. Huang, J. Chen, L. Huang, Q. Zhu, Dynamic games for secure and resilient control system design. Natl. Sci. Rev. nwz218 (2020)
27. Q. Zhu, T. Başar, A dynamic game-theoretic approach to resilient control system design for cascading failures, in *Proceedings of the 1st International Conference on High Confidence Networked Systems* (2012), pp. 41–46
28. T. Başar, G.J. Olsder, *Dynamic Noncooperative Game Theory*, vol. 23 (Siam, Philadelphia, 1999)
29. B. Whaley, *Practise to Deceive: Learning Curves of Military Deception Planners* (Naval Institute Press, Annapolis, 2016)
30. J.B. Bell, B. Whaley, *Cheating and Deception* (Routledge, New York, 2017)
31. R. Godson, J.J. Wirtz, *Strategic Denial and Deception: the Twenty-First Century Challenge* (Transaction Publishers, Piscataway, 2011)
32. S. Bodmer, M. Kilger, G. Carpenter, J. Jones, *Reverse Deception: Organized Cyber Threat Counter-Exploitation* (McGraw Hill Professional, New York, 2012)
33. C.F. Bond Jr., B.M. DePaulo, Individual differences in judging deception: accuracy and bias. Psychol. Bull. **134**(4), 477 (2008)
34. A. Vrij, S.A. Mann, R.P. Fisher, S. Leal, R. Milne, R. Bull, Increasing cognitive load to facilitate lie detection: the benefit of recalling an event in reverse order. Law Hum. Behav. **32**(3), 253–265 (2008)
35. R.E. Geiselman, The cognitive interview for suspects (CIS). Am. Coll. Forensic Psychol. **30**(3), 1–16 (2012)
36. D.T. Lykken, The GSR in the detection of guilt. J. Appl. Psychol. **43**(6), 385 (1959)
37. D.C. Howe, H. Nissenbaum, TrackMeNot: resisting surveillance in web search, *Lessons from the Identity Trail: Anonymity, Privacy, and Identity in a Networked Society*, vol. 23 (2009), pp. 417–436
38. V.P. Crawford, J. Sobel, Strategic information transmission. Econom.: J. Econom. Soc. 1431–1451 (1982)
39. N. Kartik, Strategic communication with lying costs. Rev. Econ. Stud. **76**(4), 1359–1395 (2009)
40. S. Astyk, A. Newton, C.F. Camerer, Pinocchio's pupil: using eyetracking and pupil dilation to understand truth telling and deception in sender-receiver games. Am. Econ. Rev. **100**(3), 984–1007 (2010)
41. U. Gneezy, Deception: the role of consequences. Am. Econ. Rev. **95**(1), 384–394 (2005)

42. S. Hurkens, N. Kartik, Would I lie to you? On social preferences and lying aversion. Exp. Econ. **12**(2), 180–192 (2009)
43. U. Fischbacher, F. Föllmi-Heusi, Lies in disguise—an experimental study on cheating. J. Eur. Econ. Assoc. **11**(3), 525–547 (2013)
44. G.A. Akerlof, R.J. Shiller, *Phishing for Phools: the Economics of Manipulation and Deception* (Princeton University Press, Princeton, 2015)
45. L.J. Janczewski, A.M. Colarik, *Cyber Warfare and Cyber Terrorism* (Information Science Reference, New York, 2008)
46. U.F. Minhas, J. Zhang, T. Tran, R. Cohen, A multifaceted approach to modeling agent trust for effective communication in the application of mobile ad hoc vehicle networks. IEEE Trans. Syst. Man Cybern. Part C (Appl. Rev.) **41**, 407–420 (2011)
47. Department of defense strategy for operating in cyberspace. Technical report, U.S. Department of Defense (2011)
48. F.J. Stech, K.E. Heckman, B.E. Strom, Integrating cyber D&D into adversary modeling for active cyber defense, *Cyber Deception* (Springer, Berlin, 2016), pp. 169–201
49. Q. Zhu, T. Başar, Game-theoretic approach to feedback-driven multi-stage moving target defense, *Decision and Game Theory for Security* (Springer, Berlin, 2013), pp. 246–263
50. T.E. Carroll, D. Grosu, A game theoretic investigation of deception in network security. Secur. Commun. Netw. **4**(10), 1162–1172 (2011)
51. N. Zhang, W. Yu, X. Fu, S.K. Das, gPath: a game-theoretic path selection algorithm to protect Tor's anonymity, *Decision and Game Theory for Security* (Springer, Berlin, 2010), pp. 58–71
52. Y. Sun, H. Song, A.J. Jara, R. Bie, Internet of Things and big data analytics for smart and connected communities. IEEE Access **4**, 766–773 (2016)
53. IoT news network: updates from the Internet of Things. IoT News Network, http://www.iotnewsnetwork.com/body-health/blufit-the-smart-water-bottle/
54. Q. Zhu, T. Basar, Game-theoretic methods for robustness, security, and resilience of cyber-physical control systems: games-in-games principle for optimal cross-layer resilient control systems. IEEE Control Syst. Mag. **35**(1), 46–65 (2015)
55. J. Chen, Q. Zhu, Security investment under cognitive constraints: a gestalt Nash equilibrium approach, in *2018 52nd Annual Conference on Information Sciences and Systems (CISS)* (IEEE, 2018), pp. 1–6
56. J. Chen, Q. Zhu, Control of multi-layer mobile autonomous systems in adversarial environments: a games-in-games approach. IEEE Trans. Control Netw. Syst. (2019)
57. J.B. Graves, A. Acquisti, N. Christin, Should payment card issuers reissue cards in response to a data breach? in *Workshop on the Economics of Information Security* (2014)
58. R.B. Myerson, *Game Theory: Analysis of Conflict* (Harvard University Press, Cambridge, 1991)
59. T. Alpcan, T. Basar, A game theoretic approach to decision and analysis in network intrusion detection, in *IEEE Conference on Decision and Control*, vol. 3 (IEEE, 2003), pp. 2595–2600
60. Q. Zhu, H. Tembine, T. Başar, Network security configurations: a nonzero-sum stochastic game approach, in *Proceedings of the 2010 American Control Conference* (IEEE, 2010), pp. 1059–1064
61. Q. Zhu, T. Başar, Dynamic policy-based IDS configuration, in *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) Held Jointly with 2009 28th Chinese Control Conference* (IEEE, 2009), pp. 8600–8605
62. R. Zhang, Q. Zhu, Secure and resilient distributed machine learning under adversarial environments, in *IEEE Information Fusion* (2015), pp. 644–651
63. W. Wang, Q. Zhu, On the detection of adversarial attacks against deep neural networks, in *Proceedings of the 2017 Workshop on Automated Decision Making for Active Cyber Defense* (2017), pp. 27–30
64. R. Zhang, Q. Zhu, A game-theoretic defense against data poisoning attacks in distributed support vector machines, in *2017 IEEE 56th Annual Conference on Decision and Control (CDC)* (IEEE, 2017), pp. 4582–4587

65. R. Zhang, Q. Zhu, Game-theoretic defense of adversarial distributed support vector machines. J. Adv. Inf. Fusion **14**(1), 3–21 (2019)
66. T. Basar, The Gaussian test channel with an intelligent jammer. IEEE Trans. Inform. Theory **29**(1), 152–157 (1983)
67. Q. Zhu, W. Saad, Z. Han, H.V. Poor, T. Başar, Eavesdropping and jamming in next-generation wireless networks: a game-theoretic approach, in *2011-MILCOM 2011 Military Communications Conference* (IEEE, 2011), pp. 119–124
68. Z. Xu, Q. Zhu, A game-theoretic approach to secure control of communication-based train control systems under jamming attacks, in *Proceedings of the ACM International Workshop on Safe Control of Connected and Autonomous Vehicles* (2017), pp. 27–34
69. Y. Nugraha, T. Hayakawa, A. Cetinkaya, H. Ishii, Q. Zhu, Subgame perfect equilibrium analysis for jamming attacks on resilient graphs, in *2019 American Control Conference (ACC)* (IEEE, 2019), pp. 2060–2065
70. L. Huang, J. Chen, Q. Zhu, A factored MDP approach to optimal mechanism design for resilient large-scale interdependent critical infrastructures, in *2017 Workshop on Modeling and Simulation of Cyber-Physical Energy Systems (MSCPES)* (IEEE, 2017), pp. 1–6
71. L. Huang, Q. Zhu, A dynamic games approach to proactive defense strategies against advanced persistent threats in cyber-physical systems. Comput. Secur. **89**, 101660 (2020)
72. L. Huang, J. Chen, Q. Zhu, A large-scale Markov game approach to dynamic protection of interdependent infrastructure networks, in *International Conference on Decision and Game Theory for Security* (Springer, 2017), pp. 357–376
73. J. Chen, Q. Zhu, *A Game-and Decision-Theoretic Approach to Resilient Interdependent Network Analysis and Design* (Springer, Berlin, 2020)
74. Q. Zhu, T. Başar, Robust and resilient control design for cyber-physical systems with an application to power systems, in *2011 50th IEEE Conference on Decision and Control and European Control Conference* (IEEE, 2011), pp. 4066–4071
75. F. Miao, Q. Zhu, M. Pajic, G.J. Pappas, A hybrid stochastic game for secure control of cyber-physical systems. Automatica **93**, 55–63 (2018)
76. F. Miao, Q. Zhu, A moving-horizon hybrid stochastic game for secure control of cyber-physical systems, in *2014 IEEE 53rd Annual Conference on Decision and Control (CDC)* (IEEE, 2014), pp. 517–522
77. C. Rieger, I. Ray, Q. Zhu, M.A. Haney (eds.), *Industrial Control Systems Security and Resiliency: Practice and Theory*, vol. 75 (Springer Nature, New York, 2019)
78. J. Pita, M. Jain, J. Marecki, F. Ordóñez, C. Portway, M. Tambe, C. Western, P. Paruchuri, S. Kraus, Deployed ARMOR protection: the application of a game theoretic model for security at the Los Angeles international airport, in *Proceedings of the International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track* (2008), pp. 125–132
79. E. Shieh, B. An, R. Yang, M. Tambe, C. Baldwin, J. DiRenzo, B. Maule, G. Meyer, PROTECT: a deployed game theoretic system to protect the ports of the United States, in *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems-Volume 1* (International Foundation for Autonomous Agents and Multiagent Systems, 2012), pp. 13–20
80. J.F. Nash, Equilibrium points in n-person games. Proc. Natl. Acad. Sci. USA **36**(1), 48–49 (1950)
81. H. Von Stackelberg, *Marktform und Gleichgewicht* (Springer, Berlin, 1934)
82. M. Maschler, E. Solan, S. Zamir, *Game Theory (Translated from the Hebrew by Ziv Hellman and edited by Mike Borns)* (Cambridge University Press, Cambridge, 2013)
83. H. Kuhn, Extensive games and the problem of information, in *Contributions to the Theory of Games*, ed. by H. Kuhn, A. Tucker (2016), pp. 193–216
84. J.C. Harsanyi, Games with incomplete information played by "Bayesian" players, i–iii part i. The basic model. Manag. Sci. **14**(3), 159–182 (1967)
85. J.C. Harsanyi, Games with incomplete information played by "Bayesian" players part ii. Bayesian equilibrium points. Manag. Sci. **14**(5), 320–334 (1968)

86. J.C. Harsanyi, Games with incomplete information played by “Bayesian" players, part iii. The basic probability distribution of the game. Manag. Sci. **14**(7), 486–502 (1968)
87. R.B. Myerson, Comments on “games with incomplete information played by “Bayesian” players, i–iii Harsanyi’s games with incomplete information”. Manag. Sci. **50**(12)_supplement, 1818–1824 (2004)
88. V. Krishna, *Auction Theory* (Academic, New York, 2009)
89. R. Gibbons, *Game Theory for Applied Economists* (Princeton University Press, Princeton, 1992)
90. D. Fudenberg, J. Tirole, *Game Theory* (Cambridge, 1991)
91. J.E. Mahon, The definition of lying and deception, in *The Stanford Encyclopedia of Philosophy*, ed. by E.N. Zalta, Winter 201 edn. (2016)
92. X. Feng, Z. Zheng, P. Mohapatra, D. Cansever, A Stackelberg game and Markov modeling of moving target defense, *Decision and Game Theory for Security* (Springer, Berlin, 2017), pp. 315–335
93. R. Shokri, Privacy games: optimal user-centric data obfuscation. Proc. Priv. Enhancing Technol. **2**, 299–315 (2015)
94. J. Pawlick, Q. Zhu, A mean-field Stackelberg game approach for obfuscation adoption in empirical risk minimization, in *IEEE Global Signal and Information Processing* (2017), pp. 518–522
95. R. Píbil, V. Lisỳ, C. Kiekintveld, B. Bošanskỳ, M. Pechoucek, Game theoretic model of strategic honeypot selection in computer networks, *Decision and Game Theory for Security* (Springer, Berlin, 2012), pp. 201–220
96. C. Kiekintveld, V. Lisỳ, R. Píbil, Game-theoretic foundations for the strategic use of honeypots in network security, *Cyber Warfare* (Springer, Berlin, 2015), pp. 81–101
97. J. Uitto, S. Rauti, S. Laurén, V. Leppänen, A survey on anti-honeypot and anti-introspection methods, in *World Conference on Information Systems and Technologies* (Springer, 2017), pp. 125–134
98. M. Dornseif, T. Holz, C.N. Klein, Nosebreak-attacking honeynets, in *Proceedings of the IEEE SMC Information Assurance Workshop* (IEEE, 2004), pp. 123–129
99. T. Holz, F. Raynal, Detecting honeypots and other suspicious environments, in *Proceedings of the IEEE SMC Information Assurance Workshop* (IEEE, 2005), pp. 29–36
100. N.C. Rowe, B.T. Duong, E.J. Custy, Fake honeypots: a defensive tactic for cyberspace, in *Proceedings of the IEEE Workshop on Information Assurance* (2006), pp. 223–230
101. J. Pawlick, E. Colbert, Q. Zhu, A game-theoretic taxonomy and survey of defensive deception for cybersecurity and privacy. ACM Comput. Surv. (CSUR) **52**(4), 1–28 (2019)
102. N. Baimukan, ConcealGAN, Github Project (2019), https://github.com/nurpeiis/Steganography-using-GANs
103. J. Yuill, M. Zappe, D. Denning, F. Feer, Honeyfiles: deceptive files for intrusion detection, in *Proceedings from the Fifth Annual IEEE SMC Information Assurance Workshop, 2004* (IEEE, 2004), pp. 116–122
104. W.A. Casey, Q. Zhu, J.A. Morales, B. Mishra, Compliance control: managed vulnerability surface in social-technological systems via signaling games, in *Proceedings of the 7th ACM CCS International Workshop on Managing Insider Security Threats* (2015), pp. 53–62
105. L. Spitzner, The honeynet project: trapping the hackers. IEEE Secur. Priv. **99**(2), 15–23 (2003)
106. A. Greenberg, Apple’s ‘differential privacy’ is about collecting your data—but not your data. WIRED (2016), https://www.wired.com/2016/06/apples-differential-privacy-collecting-data/
107. N. Statt, Google is open-sourcing a tool for data scientists to help protect private information. The Verge (2019), https://www.theverge.com/2019/9/5/20850465/google-differential-privacy-open-source-tool-privacy-data-sharing
108. J. Freudiger, M.H. Manshaei, J.-P. Hubaux, D.C. Parkes, On non-cooperative location privacy: a game-theoretic analysis, in *Proceedings of the ACM Conference on Computer and Communications Security* (ACM, 2009), pp. 324–337

109. J. Zhuang, V.M. Bier, O. Alagoz, Modeling secrecy and deception in a multiple-period attacker-defender signaling game. Eur. J. Oper. Res. **203**(2), 409–418 (2010)
110. K. Durkota, V. Lisỳ, B. Bošanský, C. Kiekintveld, Optimal network security hardening using attack graph games, in *International Joint Conference on Artificial Intelligence* (2015), pp. 526–532
111. K. Horák, Q. Zhu, B. Bošanský, Manipulating adversary's belief: a dynamic game approach to deception by design in network security, *Decision and Game Theory for Security* (Springer, Berlin, 2017), pp. 273–294
112. J. Filar, K. Vrieze, *Competitive Markov Decision Processes* (Springer Science & Business Media, New York, 2012)
113. MITRE, Science of cyber-security (2010)
114. NISO, Guidelines for the construction, format, and management of monolingual controlled vocabularies (2005)
115. H. Chisholm, Predicables, *Encyclopedia Britannica*, 11th edn. (Cambridge University Press, Cambridge, 1911)
116. H. Cott, *Adaptive Coloration in Animals* (Methuen, London, 1940)
117. S. Roy, C. Ellis, S. Shiva, D. Dasgupta, V. Shandilya, Q. Wu, A survey of game theory as applied to network security, in *IEEE International Conference on System Sciences* (2010), pp. 1–10
118. C.T. Do, N.H. Tran, C. Hong, C.A. Kamhoua, K.A. Kwiat, E. Blasch, S. Ren, N. Pissinou, S.S. Iyengar, Game theory for cyber security and privacy. ACM Comput. Surv. (CSUR) **50**(2), 30 (2017)
119. H. Rothstein, B. Whaley, *The Art and Science of Military Deception* (Artech House, Boston, 2013)
120. M. Bennett, E. Waltz, *Counterdeception Principles and Applications for National Security* (Artech House, Boston, 2007)
121. N.C. Rowe, J. Rrushi, *Introduction to Cyberdeception* (Springer, Berlin, 2016)
122. K.E. Heckman, F.J. Stech, R.K. Thomas, B. Schmoker, A.W. Tsow, *Cyber Denial, Deception and Counter Deception* (Springer, Berlin, 2015)
123. A. Oltramari, L.F. Cranor, R.J. Walls, P.D. McDaniel, Building an ontology of cyber security, in *Conference on Semantic Technology for Intelligence, Defense, and Security (STIDS)* (2014), pp. 54–61
124. N.C. Rowe, A taxonomy of deception in cyberspace, in *International Conference on Information Warfare and Security* (2006)
125. F. Brunton, H. Nissenbaum, *Obfuscation: A User's Guide for Privacy and Protest* (MIT Press, Cambridge, 2015)
126. B. Grosser, Privacy through visibility: disrupting NSA surveillance with algorithmically generated "Scary" stories. University of Wisconsin-Milwaukee (2014)
127. P. Chan, R. Sircar, Bertrand and Cournot mean field games. Appl. Math. Optim. **71**(3), 533–569 (2015)
128. K. Chaudhuri, C. Monteleoni, A.D. Sarwate, Differentially private empirical risk minimization. J. Mach. Learn. Res. **12**, 1069–1109 (2011)
129. C. Dwork, Differential privacy, *Automata, Languages and Programming* (Springer, Berlin, 2006), pp. 1–12
130. C. Dwork, A. Roth, The algorithmic foundations of differential privacy. Found. Trends Theor. Comput. Sci. **9**(3–4), 211–407 (2014)
131. A. Ghosh, A. Roth, Selling privacy at auction, *Games and Economic Behavior* (2015), pp. 334–346
132. D. Xiao, Is privacy compatible with truthfulness? in *Proceedings of the ACM Conference on Innovations in Theoretical Computer Science* (2013), pp. 67–86
133. M. Chessa, J. Grossklags, P. Loiseau, A short paper on the incentives to share private information for population estimates, *Financial Cryptography and Data Security* (Springer, Berlin, 2015), pp. 427–436

134. M. Chessa, J. Grossklags, P. Loiseau, A game-theoretic study on non-monetary incentives in data analytics projects with privacy implications, in *IEEE Computer Security Foundations Symposium (CSF)* (2015), pp. 90–104
135. J. Pawlick, Q. Zhu, Deception by design: evidence-based signaling games for network defense, in *Workshop on the Economics of Information Security* (Delft, The Netherlands, 2015)
136. J. Pawlick, S. Farhang, Q. Zhu, Flip the cloud: cyber-physical signaling games in the presence of advanced persistent threats, *Decision and Game Theory for Security* (Springer, Berlin, 2015), pp. 289–308
137. C. Silverman, This analysis shows how viral fake election news stories outperformed real news on Facebook. Buzzfeed news, https://www.buzzfeed.com/craigsilverman/
138. M. Ott, Y. Choi, C. Cardie, J.T. Hancock, Finding deceptive opinion spam by any stretch of the imagination, in *Proceedings of the Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1* (2011), pp. 309–319
139. V.P. Crawford, Lying for strategic advantage: rational and boundedly rational misrepresentation of intentions. Am. Econ. Rev. **93**(1), 133–149 (2003)
140. J. Pawlick, E. Colbert, Q. Zhu, Modeling and analysis of leaky deception using signaling games with evidence. IEEE Trans. Inf. Forensics Secur. **14**(7), 1871–1886 (2019)
141. W.J. Youden, Index for rating diagnostic tests. Cancer **3**(1), 32–35 (1950)
142. P. Chen, L. Desmet, C. Huygens, A study on advanced persistent threats, in *IFIP International Conference on Communications and Multimedia Security* (Springer, 2014), pp. 63–72
143. P.M. Jones, C.M. Mitchell, Human-computer cooperative problem solving: theory, design, and evaluation of an intelligent associate system. IEEE Trans. Syst. Man Cybern. **25**, 1039–1053 (1995)
144. P.R. Milgrom, Good news and bad news: representation theorems and applications. Bell J. Econ. 380–391 (1981)
145. S.J. Grossman, The informational role of warranties and private disclosure about product quality. J. Law Econ. **24**(3), 461–483 (1981)
146. S.J. Grossman, O.D. Hart, Disclosure laws and takeover bids. J. Financ. **35**(2), 323–334 (1980)
147. E. Akyol, C. Langbort, T. Başar, Information-theoretic approach to strategic communication as a hierarchical game. Proc. IEEE **105**(2), 205–218 (2017)
148. B.C. Levy, Binary and M-ary hypothesis testing, *Principles of Signal Detection and Parameter Estimation* (Springer Science & Business Media, New York, 2008)
149. R. Avenhaus, B. Von Stengel, S. Zamir, Inspection games, *Handbook of Game Theory with Economic Applications*, vol. 3 (2002), pp. 1947–1987
150. M. Albanese, E. Battista, S. Jajodia, Deceiving attackers by creating a virtual attack surface, *Cyber Deception* (Springer, Berlin, 2016), pp. 169–201
151. National vulnerability database, https://nvd.nist.gov/. Accessed Apr 2019
152. L. Wang, S. Jajodia, A. Singhal, P. Cheng, S. Noel, k-zero day safety: a network security metric for measuring the risk of unknown vulnerabilities. IEEE Trans. Dependable Secur. Comput. **11**(1), 30–44 (2014)
153. R. Bellman, On the theory of dynamic programming. Proc. Natl. Acad. Sci. **38**(8), 716–719 (1952)
154. MATLAB, R2017b. Natick, Massachusetts: The MathWorks Inc. (2017)
155. J. Pawlick, Q. Zhu, Quantitative models of imperfect deception in network security using signaling games with evidence. IEEE Commun. Netw. Secur. 394–395 (2017)
156. M.A. Noureddine, A. Fawaz, W.H. Sanders, T. Başar, A game-theoretic approach to respond to attacker lateral movement, *Decision and Game Theory for Security* (Springer, Berlin, 2016), pp. 294–313
157. J. Pawlick, J. Chen, Q. Zhu, iSTRICT: an interdependent strategic trust mechanism for the cloud-enabled internet of controlled things. IEEE Trans. Inf. Forensics Secur. **14**(6), 1654–1669 (2019)
158. M. van Dijk, A. Juels, A. Oprea, R.L. Rivest, FlipIt: the game of "Stealthy Takeover". J. Cryptol. **26**(4), 655–713 (2013)

159. Cyber Physical Systems Vision Statement. Technical report, Networking and Information Technology Research and Development Program (2015)
160. Cloud security report. Technical report, Alert logic (2015)
161. E. Fernandes, J. Paupore, A. Rahmati, D. Simionato, M. Conti, A. Prakash, FlowFence: practical data protection for emerging IoT application frameworks, in *25th USENIX Security Symposium*, pp. 531–548
162. K.D. Bowers, M. Van Dijk, R. Griffin, A. Juels, A. Oprea, R.L. Rivest, N. Triandopoulos, Defending against the unknown enemy: applying FlipIt to system security, *Decision and Game Theory for Security* (Springer, Berlin, 2012), pp. 248–263
163. K. Baumgartner, M. Golovkin, The Naikon APT: tracking down geo-political intelligence across APAC one nation at a time, https://securelist.com/analysis/publications/69953/the-naikon-apt/
164. B.J. Fogg, H. Tseng, The elements of computer credibility, in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (ACM, 1999), pp. 80–87
165. Z. Yan, P. Zhang, A.V. Vasilakos, A survey on trust management for Internet of Things. J. Netw. Comput. Appl. **42**, 120–134 (2014)
166. U. D. of Justice, Manhattan U.S. attorney announces charges against seven Iranians for conducting coordinated campaign of cyber attacks against U.S. financial sector on behalf of Islamic revolutionary guard corps-sponsored entities, https://www.justice.gov/
167. S. Siadat, A.M. Rahmani, H. Navid, Identifying fake feedback in cloud trust management systems using feedback evaluation component and Bayesian game model. J. Supercomput. **73**(6), 2682–2704 (2017)
168. T.H. Noor, Q.Z. Sheng, A. Alfazi, Reputation attacks detection for effective trust assessment among cloud services, in *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)* (2013), pp. 469–476
169. G.F. Franklin, J.D. Powell, M.L. Workman, *Digital Control of Dynamic Systems*, vol. 3 (Addison-Wesley, Menlo Park, 1998)
170. I.-K. Cho, D.M. Kreps, Signaling games and stable equilibria. Q. J. Econ. 179–221 (1987)
171. S. Kakutani, A generalization of Brouwer's fixed point theorem. Duke Math. J. **8**(3), 457–459 (1941)
172. E. Guizzo, How Google's self-driving car works. IEEE Spectr. Online **18** (2011)
173. O. Levander, Forget autonomous cars—autonomous ships are almost here. IEEE Spectr. (2017)
174. C. Zhang, J.M. Kovacs, The application of small unmanned aerial systems for precision agriculture: a review. Precis. Agric. **13**(6), 693–712 (2012)
175. D. Wakabayashi, Uber's self-driving cars were struggling before Arizona crash. The New York Times (2018)
176. S. Shane, Sleeping sailors on the U.S.S. Fitzgerald awoke to a calamity at sea. The New York Times (2017)
177. Z. Xu, Q. Zhu, Secure and resilient control design for cloud enabled networked control systems, in *Proceedings of the ACM Workshop on Cyber-Physical Systems-Security and/or Privacy* (2015), pp. 31–42
178. K.J. Aström, R.M. Murray, *Feedback Systems: an Introduction for Scientists and Engineers* (Princeton University Press, Princeton, 2010)
179. MATLAB, R2015b. Natick, Massachusetts: The MathWorks Inc. (2015)
180. A. Laszka, G. Horvath, M. Felegyhazi, L. Buttyán, FlipThem: modeling targeted attacks with FlipIt for multiple resources, *Decision and Game Theory for Security* (Springer, Berlin, 2014), pp. 175–194
181. J. Chen, Q. Zhu, Security as a service for cloud-enabled internet of controlled things under advanced persistent threats: a contract design approach. IEEE Trans. Inf. Forensics Secur. **12**(11), 2736–2750 (2017)
182. A. Cenedese, A. Zanella, L. Vangelista, M. Zorzi, Padova smart city: an urban internet of things experimentation, in *IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Nets (WoWMoM)* (IEEE, 2014), pp. 1–6

183. P. Zhang, Y. Kong, M. Zhou, A domain partition-based trust model for unreliable clouds. IEEE Trans. Inf. Forensics Secur. **13**(9), 2167–2178 (2018)
184. C. Zhu, H. Nicanfar, V.C. Leung, L.T. Yang, An authenticated trust and reputation calculation and management system for cloud and sensor networks integration. IEEE Trans. Inf. Forensics Secur. **10**(1), 118–131 (2015)
185. W. Fan, S. Yang, J. Pei, A novel two-stage model for cloud service trustworthiness evaluation. Expert Syst. **31**(2), 136–153 (2014)
186. Free community-based mapping, traffic and navigation app. Waze Mobile, https://www.waze.com/
187. R.J. Radke, T.-K. Woodstock, M.H. Imam, A.C. Sanderson, S. Mishra, Advanced sensing and control in the smart conference room at the center for lighting enabled systems and applications, in *SID Symposium Digest of Technical Papers*, vol. 47 (Wiley Online Library, 2016), pp. 193–196
188. T. Byers, Demand response and the IoT: using data to maximize customer benefit. Comverge Blog (2017), http://www.comverge.com/blog/february-2017/demand-response-and-iot-using-data-to-maximize-cus/
189. B. Herzberg, D. Bekerman, I. Zeifman, Breaking down Mirai: an IoT DDoS botnet analysis, Incapsula Blog, Bots and DDoS, Security (2016), https://www.incapsula.com/blog/malware-analysis-mirai-ddos-botnet.html
190. D. Lewis, *Convention: a Philosophical Study* (Wiley, New York, 2008)
191. R.B. Myerson, Population uncertainty and Poisson games. Int. J. Game Theory **27**(3), 375–392 (1998)
192. Account lockout threshold. Microsoft TechNet (2014), https://technet.microsoft.com/en-us/library/hh994574(v=ws.11).aspx
193. D.J. Hammerstrom, Part II. Grid friendly appliance project, in *GridWise Testbed Demonstration Projects* (Pacific Northwest National Laboratory, 2007)
194. J.D. Glover, M.S. Sarma, T. Overbye, *Power System Analysis and Design, SI Version* (Cengage Learning, Boston, 2012)
195. A. Bensoussan, M. Kantarcioglu, S.C. Hoe, A game-theoretical approach for finding optimal strategies in a botnet defense model, *Decision and Game Theory for Security* (Springer, Berlin, 2010), pp. 135–148
196. K.J. Higgins, Conficker botnet 'dead in the water,' researcher says. Dark Reading (2010), http://www.darkreading.com/attacks-breaches/conficker-botnet-dead-in-the-water-researcher-says/d/d-id/1133327?
197. A. Mohammadi, M.H. Manshaei, M.M. Moghaddam, Q. Zhu, A game-theoretic analysis of deception over social networks using fake avatars, *Decision and Game Theory for Security* (Springer, Berlin, 2016), pp. 382–394
198. Y. Hayel, Q. Zhu, Dynamics of strategic protection against virus propagation in heterogeneous complex networks, *Decision and Game Theory for Security* (Springer, Berlin, 2017), pp. 506–518
199. Q. Wu, S. Shiva, S. Roy, C. Ellis, V. Datla, On modeling and simulation of game theory-based defense mechanisms against DoS and DDoS attacks, in *Proceedings of the Spring Simulation Multiconference* (Society for Computer Simulation International, 2010), p. 159
200. A.-H. Mohsenian-Rad, A. Leon-Garcia, Distributed internet-based load altering attacks against smart power grids. IEEE Trans. Smart Grid **2**(4), 667–674 (2011)
201. S. Amini, H. Mohsenian-Rad, F. Pasqualetti, Dynamic load altering attacks in smart grid, in *IEEE Innovative Smart Grid Technologies Conference (ISGT)* (2015), pp. 1–5
202. J. Pawlick, Q. Zhu, Using ethically-contrained game theory to protect our privacy, in *International Workshop on Obfuscation: Science, Technology, and Theory*, pp. 18–21 (2017), http://www.obfuscationworkshop.io
203. T. Zhang, Q. Zhu, Strategic defense against deceptive civilian GPS spoofing of unmanned aerial vehicles, *Decision and Game Theory for Security* (Springer, Berlin, 2017), pp. 213–233
204. Z. Xu, Q. Zhu, Cross-layer secure cyber-physical control system design for networked 3D printers, in *American Control Conference (ACC)* (2016), pp. 1191–1196

205. M. Jain, J. Tsai, J. Pita, C. Kiekintveld, S. Rathi, M. Tambe, F. Ordónez, Software assistants for randomized patrol planning for the LAX airport police and the federal air marshal service. Interfaces **40**(4), 267–290 (2010)
206. K. Ferguson-Walter, D. LaFon, T. Shade, Friend or faux: deception for cyber defense. J. Inf. Warf. **16**(2), 28–42 (2017)

Index

J. Pawlick and Q. Zhu, *Game Theory for Cyber Deception*,
Static & Dynamic Game Theory: Foundations & Applications,
https://doi.org/10.1007/978-3-030-66065-9