Advances in Information Security 74

Cliff Wang
Zhuo Lu *Editors*

Proactive and Dynamic Network Defense

Springer

Advances in Information Security

Volume 74

More information about this series at http://www.springer.com/series/5576

Cliff Wang • Zhuo Lu
Editors

Proactive and Dynamic Network Defense

Springer

Editors

Cliff Wang
Computing and Information Science
Division
Army Research Office
Durham, NC, USA

Zhuo Lu
Department of Electrical Engineering
University of South Florida
Tampa, FL, USA

ISSN 1568-2633
Advances in Information Security
ISBN 978-3-030-10596-9 ISBN 978-3-030-10597-6 (eBook)
https://doi.org/10.1007/978-3-030-10597-6

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

This volume partially comes out a special workshop that was dedicated to discuss the latest research topics in proactive and adaptive network defense held in Fall 2017. Although network security has been studied for the past two decades resulting in many different approaches to address various types of attacks and to enhance network resiliency, proactive and adaptive defense approaches such as moving target defense and using network deception are still relatively new. There is also a lack of formal models that can effectively capture the rich dynamics between attackers and defenders under complex settings such as having multiplayers, with multiple stages and various levels of engagement, and constrained by partial information of adversaries or network situation uncertainties. The workshop has identified the following interesting research challenges that the community is pursuing: defining fundamental models for proactive network defense, game theory-based network defense frameworks, learning-based attack and defense interactions, and proactive network defense for emerging applications (e.g., wireless, mobile, and in-vehicle system applications).

This volume is designed for better understanding of proactive and dynamic network defense that has been proposed as an important alternative cyber defense mechanism toward comprehensive network defense. We present a collection of the latest fundamental research results toward understanding proactive and dynamic network defense by top researchers in related areas. This volume includes papers that offer formal frameworks to define proactive and dynamic network defense and develop novel models to analyze and evaluate proactive designs and strategies in computer systems, network systems, cyber-physical systems, and wireless networks. A wide variety of scientific techniques have been highlighted to study these problems in the fundamental domain. We sincerely hope that this volume can inspire researchers to face current research challenges and further explore more solid and rigorous scientific foundations for proactive and dynamic network defense.

Durham, NC, USA — Cliff Wang
Tampa, FL, USA — Zhuo Lu

Acknowledgments

We are extremely grateful to the contributors and authors for their great contributions to this book. We also would like to thank Susan Lagerstrom-Fife, senior publishing editor at Springer, for her support of this project.

Contents

Contributors

Alvaro A. Cardenas Erik Jonsson School of Engineering, University of Texas at Dallas, Richardson, TX, USA

Baskin School of Engineering, University of California, Santa Cruz, Santa Cruz, CA, USA

Bedri A. Cetiner Utah State University, Logan, UT, USA

Lixing Chen University of Miami, Coral Gables, FL, USA

Yingying Chen Rutgers University, New Brunswick, NJ, USA

Salvatore D'Oro Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

Koorosh Firouzbakht Northeastern University, Boston, MA, USA

Ryan M. Gerdes Virginia Tech, Arlington, VA, USA

Jairo Giraldo Erik Jonsson School of Engineering, University of Texas at Dallas, Richardson, TX, USA

Yantian Hou Boise State University, Boise, ID, USA

Aris Kanellopoulos The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Xenofon Koutsoukos Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA

Aron Laszka Department of Computer Science, University of Houston, Houston, TX, USA

Ming Li The University of Arizona, Tucson, AZ, USA

Hongbo Liu Indiana University Purdue University Indianapolis, Indianapolis, IN, USA

Jian Liu Rutgers University, New Brunswick, NJ, USA

Tommaso Melodia Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

Guevara Noubir Northeastern University, Boston, MA, USA

Yanjun Pan The University of Arizona, Tucson, AZ, USA

Francesco Restuccia Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

Masoud Salehi Northeastern University, Boston, MA, USA

Ness Shroff The Ohio State University, Columbus, OH, USA

Kyriakos G. Vamvoudakis The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA

Yevgeniy Vorobeychik Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA

Sinong Wang The Ohio State University, Columbus, OH, USA

Yan Wang Binghamton University, Binghamton, NY, USA

Jie Xu University of Miami, Coral Gables, FL, USA

Shouhuai Xu Laboratory for Cybersecurity Dynamics, Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA

Liyang Zhang Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA

Chapter 1
Cybersecurity Dynamics: A Foundation for the Science of Cybersecurity

Shouhuai Xu

Abstract Cybersecurity Dynamics is new concept that aims to achieve the modeling, analysis, quantification, and management of cybersecurity from a holistic perspective, rather than from a building-blocks perspective. It is centered at modeling and analyzing the attack-defense interactions in cyberspace, which cause a “natural” phenomenon—the evolution of the global cybersecurity state. In this chapter, we systematically introduce and review the Cybersecurity Dynamics foundation for the Science of Cybersecurity. We review the core concepts, technical approaches, research axes, and results that have been obtained in this endeavor. We outline a research roadmap towards the ultimate research goal, and identified technical barriers that poses challenges to reach the goal.

1.1 Introduction

The fundamental concepts of confidentiality, integrity, and availability have been at the core of information security research over the past decades. These concepts have led to the development of many building-block techniques, such as cryptographic mechanisms, which can be rigorously analyzed in a sound scientific framework. This motivated us to seek fundamental concepts and frameworks that can guide our investigation of cybersecurity, which has to be understood from a holistic perspective (i.e., by treating a network of interest as a whole, rather than investigating their building-blocks separately).

In the course of our endeavor, the concept of *cybersecurity dynamics* emerges [121]. Intuitively, the concept of cybersecurity dynamics reflects the *evolution* of the global cybersecurity state of a network, where “evolution” is caused by the interactions between the human parties involved—dubbed *attack-defense interactions*.

S. Xu (✉)
Laboratory for Cybersecurity Dynamics, Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA
e-mail: shxu@cs.utsa.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_1

The human parties involved include attackers who wage attacks against a network, defenders who employ defense mechanisms to protect a network in question, and users who may be exploited by the attackers to wage attacks.

The concept of Cybersecurity Dynamics is appealing because of the following. First, the global cybersecurity state of a network reflects the real-time situation, which “naturally” evolves over time because of the attack-defense interactions. Knowing the real-time global cybersecurity state or situation is of high interest to cyber defense decision-makers, who often need to adjust their defense posture (including policies, architectures, and mechanisms) to mitigate or minimize the damage of cyber attacks. Second, the effects of employing new cyber defense postures are reflected by the resulting global cybersecurity state. This means that we can compare the effectiveness of one defense posture against another. Third, looking at the evolution of the global cybersecurity state allows us to build systematic models with *descriptive* power (i.e., characterizing what phenomenon can happen under what circumstances), *prescriptive* power (i.e., guiding the adjustment to defense postures to mitigate or minimize the damage of cyber attacks), and *predictive* power (i.e., forecasting what will happen with or without making adjustments to the defense posture). Four, modeling the evolution of the global cybersecurity state makes security quantification an inherent task, which paves the way for quantitative decision-making in the course of cyber defense operations. In particular, the concept of Cybersecurity Dynamics naturally leads to the notion of *macroscopic cybersecurity*, with models that will use parameters to describe or represent (among other things) attacks and defenses.

Our Contributions The present chapter systematically refines and extends an earlier treatment of the Cybersecurity Dynamics foundation given in [119], while accommodating the many advancements that have been made during the past few years. More specifically, we systematically introduce and review the Cybersecurity Dynamics foundation (or framework), while focusing on three orthogonal, coherent “axes”: (1) the cybersecurity metrics axis aims to develop a systematic set of metrics that can adequately describe cybersecurity; (2) the cybersecurity first-principle modeling and analysis axis aims to establish cybersecurity laws governing the evolution of the global cybersecurity state; and (3) the cybersecurity data analytics axis aims to extract model parameters and validate/invalidate models developed in the first-principle modeling and analysis axis. In particular, we discuss the deep connections between these three axes. Despite the many efforts and significant results, there are many outstanding problems that have yet to be tackled. We hope the present chapter will inspire many more studies to address the many open problems.

Chapter Outline The chapter is organized as follows. Section 1.2 presents an overview of the Cybersecurity Dynamics foundation. Section 1.3 reviews the recent advancement in cybersecurity metrics research. Section 1.4 reviews the recent advancement in cybersecurity first-principle modeling and analysis. Section 1.5 reviews the recent advancement in cybersecurity data analytics. Section 1.6 discusses future research directions, including technical barriers that need to be tackled. Section 1.7 reviews related prior studies. Section 1.8 concludes the present chapter.

1.2 Overview of the Cybersecurity Dynamics Foundation

1.2.1 Terminology

By “network” we mean an arbitrary (cyber, cyber-physical, Internet of Things or IoT) network of interest that is enabled or interconnected by the TCP/IP technology, regardless of the underlying communication being wired or wireless. A network can have an arbitrarily large size (e.g., an enterprise network or even the entire cyberspace). By “computer” we mean a computer or device (e.g., smartphones, IoT devices) with a software stack, which typically includes some applications, library functions, and an operating system.

A network is protected by some *defenders*, who may or may not be under the same administrative jurisdiction (e.g., a network of interest consisting of multiple independently managed enterprise networks). Each network has a number of *users*, who are often subject to attacks (e.g., social-engineering attacks). The *attacker* attempts to compromise the computers in a network, by exploiting weaknesses in the network software and hardware as well as weaknesses in the users or defenders (e.g., making them become insider threats).

In the context of the present chapter, the terms *cybersecurity* and *security* are used interchangeably. In order to model cybersecurity from a holistic perspective (in contrast to building-block perspectives), we need to have the notion of *model resolution*, reflecting the level of abstraction. For example, we can treat a computer or software component as an indivisible unit, dubbed “*atoms*” of a model. Throughout the chapter, we will use the term “atom” to indicate the unit from a modeling point of view. Because each “atom” will be represented as a vertex or node in a graph-theoretic model, we also call an “atom” a *node*. When we treat a computer as a unit or “atom”, we are dealing with a coarse-grained model because the internal components of the computer are treated as transparent. As a consequence, compromise of any program in the user space of a computer would force us to treat the entire computer as compromised. When we treat a software component (e.g., software program or even program function) as an “atom”, we are dealing with a fine-grained model because the compromise of one component in a computer (e.g., application) does not necessarily mean the compromise of another component in the same computer (e.g., the operating system).

For each “atom” mentioned above, we can define its *security state*, which can be either *secure* but possibly vulnerable to attacks because it contain some vulnerabilities, or *compromised*. In the real world, the security state of an “atom” is dynamic (i.e., changing over time), rather than static, because it can become compromised (because of some attack actions), then become secure (because of some defense actions), then become compromised, and so on. This naturally leads to the view that the security state evolves. We call the security state of an “atom” a *local* cybersecurity state because it deals with an individual “atom”; we call the security state of an entire network the *global* cybersecurity state, which can be represented as a vector of the local cybersecurity states of the “atoms”.

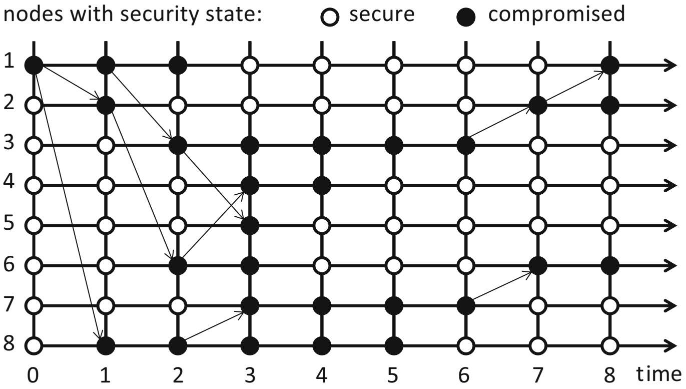


Fig. 1.1 Illustration of the evolution of the global cybersecurity state in a small network of 8 "atoms" at an appropriate model resolution. The "atoms" are represented as *nodes* (e.g., computers, devices, or software components). In the discrete-time model, each "atom" or node has a cybersecurity state at any point in time, either *secure* (represented as an empty circle) or *compromised* (represented as a filled circle) in this example. Each arrow represents a successful attack from a compromised node against a secure node, causing the latter to become compromised. A compromised node may become secure again because of some defense activities. A secure node may be attacked by multiple compromised nodes at the same time

Figure 1.1 illustrates the evolution of the global cybersecurity state of a network, reflected by the evolution of the local cybersecurity states of individual "atoms" that are represented as "nodes" 1, ..., 8. In this illustration, a node has two possible states at any point in time, *secure* (empty circle) or *compromised* (filled circle). A secure node may be attacked by one or multiple compromised nodes and then become compromised; a compromised node may become secure again because of some defense activities. An arrow indicates a successful attack.

1.2.2 Research Objectives

The evolution of the global cybersecurity state, as illustrated in Fig. 1.1, is a *natural* phenomenon in cyberspace. The core research objectives of Cybersecurity Dynamics are centered at *understanding*, *managing* (or controlling), and *forecasting* the evolution. Understanding the evolution means we want to gain deep insights into the laws that govern the evolution. For this purpose, we need to build *descriptive* models to analyze how the attack-defense interactions govern the evolution of the global cybersecurity state. Managing the evolution means that we want to mitigate or control, if not minimize, the damage so as to benefit the defender. For this purpose, we need to build *prescriptive* models that can guide the orchestration of cyber defense activities in an optimal or cost-effective fashion. Forecasting means that we want to be able to forecast or predict the evolution so as to facilitate adaptive

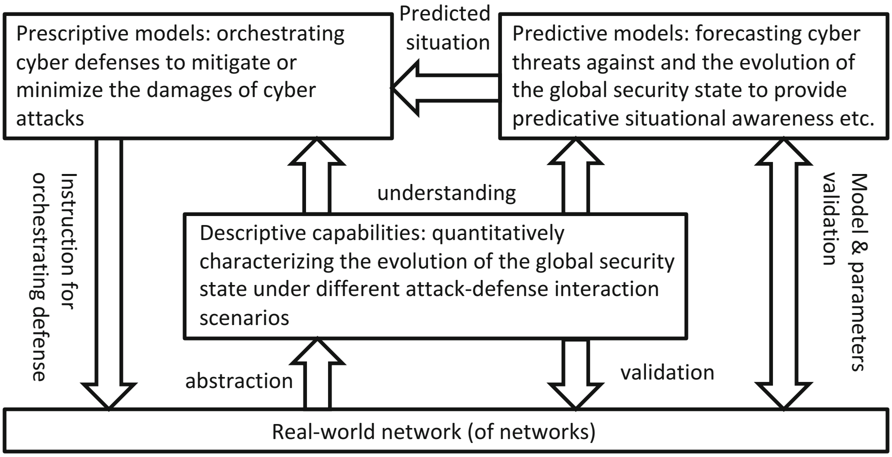

Predicted situation
Prescriptive models: orchestrating cyber defenses to mitigate or minimize the damages of cyber attacks
Predictive models: forecasting cyber threats against and the evolution of the global security state to provide predicative situational awareness etc.
Instruction for orchestrating defense
understanding
Model & parameters validation
Descriptive capabilities: quantitatively characterizing the evolution of the global security state under different attack-defense interaction scenarios
abstraction
validation
Real-world network (of networks)

Fig. 1.2 Three core research objectives of Cybersecurity Dynamics: descriptive capabilities, prescriptive capabilities, and predictive capabilities

and/or proactive cyber defense. For this purpose, we need to build *predictive* models that can forecast, among other things, the evolution of the global cybersecurity state and the incoming threats against a network of interest.

Figure 1.2 highlights the aforementioned three core research objectives and the relationship between them. Descriptive models are abstracted from the real-world networks by faithfully representing the attack-defense interactions. These models will be validated (or invalidated) according to real-world data or experiments. Predictive models are built on top of the description models and are also validated (or invalidated) according to real-world data. Prescriptive models are also built on top of descriptive models, while possibly taking into consideration the situations predicted or forecasted by the predictive models. The prescriptive models will guide the orchestration of cyber defense so as to benefit the defender in a cost-effective, if not optimal, fashion.

1.2.3 Scope

Figure 1.3 highlights the scope of the present chapter, which focuses on discussing three axes of Cybersecurity Dynamics research: (1) Cybersecurity metrics, which are driven by applications (e.g., for orchestrating cyber defenses to mitigate or minimize the damage of cyber attacks) and semantics (e.g., what aspects of cybersecurity would reflect the competence of cyber defense?). (2) Cybersecurity first-principle modeling and analysis, which are driven by assumptions. First-principle models are useful in the absence of real-world data and can be inspired by the properties exhibited by real-world datasets. (3) Cybersecurity data analytics, which are driven by real-world data or experiments.

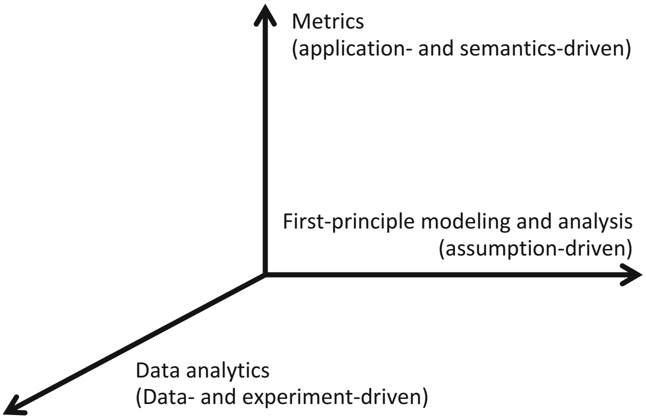

Metrics
(application- and semantics-driven)
First-principle modeling and analysis
(assumption-driven)
Data analytics
(Data- and experiment-driven)

Fig. 1.3 Scope of the present chapter: three research axes towards achieving the research objectives of Cybersecurity Dynamics

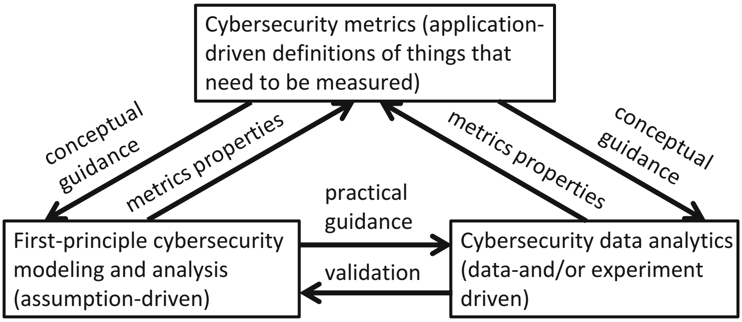

Cybersecurity metrics (application-driven definitions of things that need to be measured)
conceptual guidance
metrics properties
metrics properties
conceptual guidance
practical guidance
First-principle cybersecurity modeling and analysis (assumption-driven)
validation
Cybersecurity data analytics (data-and/or experiment driven)

Fig. 1.4 Relationship between the three research axes

Figure 1.4 highlights the relationship between the three research axes. The cybersecurity metrics axis aims to rigorously define metrics to measure and quantify cybersecurity from a holistic perspective, and therefore provides conceptual guidance to the other two axes because those quantitative models are often centered at some metrics. Along this axis, significant progress has been made [17, 18, 20, 21, 79, 89, 94, 104].

The cybersecurity first-principle modeling and analysis axis aims to build, under appropriate assumptions, mathematical models to describe the evolution of the global cybersecurity state caused by cyber attack-defense interactions. By "first-principle" we mean the use of as-simple-as-possible models with as-few-as-possible parameters, while making as-weak-as-possible assumptions; of course, these models must make sense from a cybersecurity perspective and can be validated/invalidated (e.g., through the validation/invalidation of the assumptions they make). This axis aims to establish cybersecurity laws governing the evolution of the global cybersecurity state. For example, these first-principle models aim to derive macroscopic phenomena (or characteristics or properties) from the underlying microscopic attack-defense interactions. This axis supports the cybersecurity metrics axis by providing insights into the properties of metrics (e.g., do they converge or oscillate over time), and provides practical guidance to the cybersecurity data analytics axis (e.g., by showing that some model parameters are necessary and therefore cannot be replaced with any alternatives). Along this axis, significant progress has been made [25, 40, 68, 73, 120, 123, 126, 127, 130–132, 147, 148].

The cybersecurity data analytics axis aims to use data- and/or experiment-driven studies to obtain model parameters and validate/invalidate first-principle models. This is because first-principle models typically, and legitimately, assume away the obtaining of model parameters. This axis supports the cybersecurity metrics axis by providing insights into the properties of metrics (e.g., some metrics are hard or costly to measure, suggesting the need to define and use alternate metrics), and helps validate first-principle models (e.g., by showing that an assumption underlying a first-principle model is not valid). Along this axis, significant progress has been made [15, 95, 96, 133, 134, 138–140].

1.3 Cybersecurity Metrics

The most outstanding open problem in cybersecurity research is arguably cybersecurity metrics [87, 94, 104]. Despite its clear importance, the problem is largely open as evidenced by the fact that it has been constantly listed as one of the hard problems [50, 84, 88]. Recently, the problem has received systematic attention [17, 18, 20, 21, 79, 89, 94, 104].

In Cybersecurity Dynamics [94, 119], the following five kinds of cybersecurity metrics have been proposed to systematically describe the evolution of the global cybersecurity state [94]: (1) metrics for describing a network including its configurations; (2) metrics for describing systems and human vulnerabilities; (3) metrics for describing defenses employed to protect networks; (4) metrics for describing cyber attacks (i.e., threat models); and (5) metrics for describing the global cybersecurity state or cybersecurity situational awareness.

Specifically, let *security_state*(t) denote the global cybersecurity state at time t, $C(t)$ denote a network of interest at time t (including its hardware and software configurations), $L(t)$ denote the vulnerabilities in the network at time t (including possibly zero-day vulnerabilities, human factors with uncertainty), $D(t)$ denote the defense posture at time t (i.e., the defense that are employed at time t to protect the network), and $A(t)$ denote the attacks that are waged against the network at time t. The framework aims to obtain families of mathematical functions, denoted by $\{f\}$, such that

$$\mathit{security\_state}(t) = f(C(t), L(t), D(t), A(t)). \tag{1.1}$$

Equation (1.1), once achieved, has many applications. For example, it allows us to compare the global cybersecurity of networks deploying two different configurations, say $C(t)$ vs. $C'(t)$, or two different defense postures, say $D(t)$ vs. $D'(t)$, through the difference between the corresponding evolution of $\mathit{security\_state}(t)$ and $\mathit{security\_state}'(t)$ over time. As we will discuss later, some concrete f's have been investigated in the cybersecurity first-principle modeling and analysis axis and the cybersecurity data analytics axis.

In what follows, we discuss how to obtain mathematical representations of network configurations $C(t)$, vulnerabilities $L(t)$, defense postures $D(t)$, and threats $A(t)$. These representations naturally lead to quantitative metrics.

1.3.1 Representation of Network Configuration and Metrics

Representation At a high level, configurations can be reflected by an *attack-defense structure*, which can be described as a graph $G(t) = (V(t), E(t))$, where $V(t)$ is the node or vertex set at time t, and $E(t)$ is the edge or arc set at time t. A node $v \in V(t)$ represents an "atom" mentioned above (e.g., a computer or software component). An edge or arc $(u, v) \in E(t)$ means that node u can attack node v, meaning that the communication from node u to node v may not be filtered, for example, by host-based intrusion prevention (when u and v belong to the same computer) or by network-based intrusion prevention (when u and v represent, or belong to, different computers). Moreover, $(u, v) \in E(t)$ means that the compromise of node u can cause the compromise of node v. Note that $E(t)$ does not necessarily represent the physical network topology in general (except perhaps for sensor networks or IoT networks where nodes can only afford to have short-range communications); in general, $(u, v) \in E(t)$ represents a communication link or path in a network. It turns out that filtering unauthorized communication relations $(u, v) \notin E(t)$ is an important defense means (see, for example, [17, 18, 126, 148]).

Recently, researchers have started to investigate how to represent networks at finer granularities [17, 18]. Suppose a network of interest is composed of $n(t)$ computers or devices at time t. In order to obtain the attack-defense structure $G(t) = (V(t), E(t))$, we need to first represent the *software stacks* on each computer or device, meaning that we need to model the applications, operating systems, and possibly library functions. Then, a computer or device, denoted by i, may be represent by a graph $G_i(t) = (V_i(t), E_i(t))$, where $v \in V_i$ represents an "atom" (e.g., application, operating system, or function), and $(u, v) \in E_i(t)$ means either u can call v (i.e., caller-callee dependence relation) or u can communicate with v (i.e., inter-application communication relation). Another edge set $E_0(t)$ may be defined to represent the authorized inter-computer communications within the network at time t. Yet another edge $E_*(t)$ may be defined to represent the authorized inter-network communication relations between the network and the external networks (i.e., internal-external communication relations). Note that $E_i(t)$ reflects a host-based access control policy (if employed), and $E_0(t)$ and $E_*(t)$ reflect network-wide access control policies (if employed). As a result, the attack-defense structure $G(t) = (V(t), E(t))$ may be derived as follows [17, 18]:

$$V(t) = V_1(t) \cup \ldots \cup V_{n(t)}(t) \text{ and}$$

$$E(t) = E_1(t) \cup \ldots \cup E_{n(t)}(t) \cup E_0(t) \cup E_*(t).$$

Metrics Having obtained the graph-theoretic representation $G(t) = (V(t), E(t))$, we may define metrics to characterize $G(t)$. For example, we may use nodes' degree distribution to characterize the structure of $G(t)$; we may characterize the evolution of $G(t)$ over time; we may quantify the difference of two defense policies by comparing the attack-defense structures resulting from their respective employments.

1.3.2 Representation of Vulnerabilities and Metrics

Representation We propose classifying vulnerabilities into three kinds: software, hardware, and human vulnerabilities, which are all used in a broad sense.

- We use the term "software vulnerabilities" to describe the vulnerabilities in the entire software stack, including applications, library functions, and operating systems. Software vulnerabilities are the root cause of many real-world attacks. For example, the problem of *vulnerability detection* is an active research topic (see, for example, [61, 69, 70]).
- We use the term "hardware vulnerabilities" to describe the vulnerabilities in the hardware, architecture, and firmware. The number of hardware vulnerabilities is often much smaller than the number of software vulnerabilities, but the damage caused by a hardware vulnerability is often severe because of the wide use of the hardware. Two recent examples of hardware vulnerabilities are Spectre and Meltdown (see, for example, [62, 113]).
- We use the term "human vulnerabilities" to describe the vulnerabilities of the users and administrators, such as vulnerabilities to social-engineering attacks (e.g., phishing) as well as insider threats and the vulnerabilities caused by the use of weak passwords.

Each vulnerability may be associated with a set of attributes. For example, a software vulnerability may have the following attributes: (1) the privilege that is required in order to exploit the vulnerability (e.g., local access vs. remote access); (2) what is the chance that there is a zero-day vulnerability in a software component? (3) what is the security consequence of the exploitation of a vulnerability?

Metrics Corresponding to these vulnerabilities, metrics need to be defined to quantify them. Two approaches have been proposed in the literature to measure software vulnerabilities, *coarse-grained* vs. *fine-grained*.

- Fine-grained approach: In this approach, vulnerabilities are considered at fine-grained granularities by separating the vulnerabilities of applications, library functions, and operating systems [17, 18].
- Coarse-grained approach: In this approach, vulnerabilities are often discussed at an aggregate level. For example, when treating a computer as an "atom", we consider the overall vulnerability of a computer, which can be aggregated from the vulnerabilities in the applications, library functions, and operating systems. This approach has been used in numerous cybersecurity first-principle models (see [126, 148] and the references therein).

Similarly, hardware vulnerabilities may be characterized by, for example, the chance that a vulnerability can be exploited; human vulnerabilities may be described by the chance that a user or defender is vulnerable to social-engineering attacks.

1.3.3 Representation of Defenses and Metrics

Representation There are many kinds of defense mechanisms that need to be represented for modeling purposes, such as firewalls, host-based intrusion prevention/detection systems, and network-based intrusion prevention/detection systems. Moreover, access control policies also need to be represented. For example, a tight access control policy would filter or block any unauthorized communication or function call; in contrast, a loose access control policy would not filter or block any unauthorized communication or function call, which can happen when some "atoms" are compromised. For modeling purposes, we classify defenses into preventive, reactive, proactive, adaptive, and active defenses.

- Preventive defenses aim to prevent attacks from succeeding or even reaching the target of interest. Mechanisms such as whitelisting, access control, and firewall are examples of preventive defenses.
- Reactive defenses aim to detect successful attacks and "clean up" their damage. Mechanisms such as anti-malware tools are examples of reactive defenses.
- Adaptive defenses aim to dynamically adjust the defense posture so as to mitigate or disrupt ongoing attacks that have been detected by the defender. Examples include the use of Software-Defined Networking (SDN) technology to change network configurations, or route network traffic through dynamically employed network security tools such as firewalls and intrusion prevention/detection systems. A concrete example for protecting systems with known, but unpatched, vulnerabilities is shown in [16].
- Proactive defenses aim to dynamically adjust the defense posture so as to mitigate or disrupt attacks, whose presence is not necessarily known to the defender. Mechanisms such as Moving Target Defense (MTD) are examples of proactive defenses.
- Active defenses aim to deploy defense mechanisms (or defenseware) to "patrol" networks to detect and clean up compromises. In the context of the present

Chapter, active defenses are not meant to be "hacking back" because the defenseware are deployed within the boundary of the defender's network.

Metrics Metrics need to be defined to measure the defense capabilities of a defender. For a preventive defense mechanism, we need to measure what kinds of cyber attacks that can or cannot be prevented by it. For a reactive defense mechanism, its detection capabilities can be measured by the false-positive rate, false-negative rate, and related metrics; similarly, its "cleaning" capabilities may not be perfect as well (because there is evidence showing that using multiple anti-malware tools together is not adequate to clean up malware infecting a computer [33, 80, 81, 97]). For adaptive defense, its capabilities before and after an adaptation should be different (e.g., in terms of both attack-prevention and attack-detection capabilities). For proactive defense, its capabilities can be measured by the extent to which the compromised nodes can be cleaned by such mechanisms. For active defense, its capabilities can be measured by what kinds of attacks can be detected and cleaned up by such mechanisms.

1.3.4 Representation of Attacks and Metrics

Representation There are many kinds of cyber attacks, which can be characterized from multiple perspectives. From the perspective of *attack freshness*, which often reflects the *attack evasion capability*, we can classify attacks into the following categories:

- Zero-day attacks: These attacks can be further divided into two sub-categories, depending on the freshness of the vulnerabilities they exploit.
  - Zero-day attacks exploiting zero-day vulnerabilities: These attacks exploit zero-day vulnerabilities which are not known to anyone but the attacker, the exploit writer, or the entity that discovered the vulnerability. These attacks are often difficult to detect, let alone prevent. These attacks can also accommodate the exploitation of newly compromised employees as *insider threats*.
  - Zero-day attacks exploiting known vulnerabilities: These attacks exploit known, but unpatched, vulnerabilities, while possibly able to evade any existing defense systems (e.g., intrusion prevention/detection systems).
- Known attacks: These attacks are recognized by defense systems and therefore can be blocked before they cause any damage or detected after they penetrate into computers or devices.

From the perspective of *attack behaviors*, which often reflect the characteristics of attackers, we can classify attacks into the following categories:

- Machine-waged attacks: These attacks are largely waged by machines and are largely automated.

  - Push-based attacks: These attacks actively seek to compromise other computers or devices [126]. Examples of these attacks are computer malware, which actively search for vulnerable victims. Social engineering attacks also fall into this category.
  - Pull-based attacks: These attacks passively wait to compromise other computers or devices [126]. Examples of these attacks are "drive-by download" by which a malicious web server waits for connections from vulnerable browsers and then compromises the latter [102].

- Human-waged attacks: These attacks are largely waged by human attackers and are largely manual.

  - Advanced Persistent Threats (APTs): These attacks are often waged by patient attackers targeting high-value assets. These attacks are often carefully planned.
  - Insider Threats: These attacks are largely waged by compromised users who are authorized with some privileges. These attackers are often victims of social engineering attacks, but are aware of their own malicious activities (in contrast to other victims of social engineering attacks, such as those who are lured to double-click a malicious email attachment or access a malicious website).

From the perspective of *attack objectives*, we can classify attacks into the following categories:

- Attacks against confidentiality: These attacks attempt to compromise the confidentiality of data, either during transmission, which is possible when the cryptographic protection mechanisms or protocols are flawed, or during storage in computer memory or disks, which is possible by penetrating into the computers [22, 39, 41, 91] or using side-channel attacks [63].
- Attacks against integrity: These attacks attempt to compromise the integrity of data, either during transmission, which is possible when the cryptographic protection mechanisms or protocols are flawed, or during storage in computer memory or disks, which is possible (for example) when the storage provider is malicious (see, e.g., [54, 142–146]).
- Attacks against availability: These attacks attempt to make services unavailable to their users [48]. These attacks are often waged by many compromised computers or devices, such as botnets [26, 57, 66, 141].

Faithful threat or attack models are important. For example, both random and targeted deletions of nodes from computer networks [3] oversimplifies real-world attacks [109, 114].

Metrics Many kinds of metrics can be defined to measure attack capabilities, such as (1) the *exploits* that can be used by the attacker; (2) the *agility* of the attacker, and (3) the *strategy* that can be used by the attacker.

- Characterizing exploits: An exploit can be described by its attributes, such as: whether it exploits a zero-day vulnerability or an unpatched but known vulnerability.
- Characterizing attack agility: This attribute aims to describe how active and agile the attacker is. For example, one attacker may only reactively update its exploits after the defender updates its defenses. The first study at modeling and quantifying the agility of attackers is reported in [79], which presents a metrics framework for transforming well-defined security metrics (e.g., false-positive rate and false-negative rates) to measure attacker agility.
- Attack strategies: Examples of attack strategies are Lockheed Martin's Cyber Kill Chain [49] and Mandiant's Attack Life Cycle [77]. A general attack strategy may include the following phases: reconnaissance, weaponization, initial compromise, further reconnaissance, privilege escalation, and lateral movement. At each phase, metrics need to be defined to measure the attack capabilities.

1.3.5 Security State Metrics

For any model resolution (e.g., treating a computer/device as an atom vs. treating a software component as an atom), the security state of an "atom" can be in one of multiple states, such as *secure* vs. *compromised*, denoted by

$$security\_state(atom, t)$$

$$=$$

$$\begin{cases} 0 & \text{the atom is in the } secure \text{ state at time } t \\ 1 & \text{the atom is in the } compromised \text{ state at time } t \end{cases}$$

Therefore, at any point in time, the *global* cybersecurity state can be defined as

$$global\_security(t)$$

$$= \frac{\text{the number of atoms in the } compromised \text{ state at time } t}{\text{the total number of atoms at time } t},$$

while noting that the total number of "atoms" can dynamically evolve. This is arguably one of the most fundamental metrics and has been the center of numerous cybersecurity first-principle models [119].

1.4 Cybersecurity First-Principle Modeling and Analysis

At a high level, cybersecurity first-principle modeling aims to design and characterize the various kinds of mathematical functions f illustrated in Eq. (1.1). Several kinds of f's have been proposed to describe different kinds of attack-defense interactions and the resulting dynamics [119]: preventive and reactive cyber defense dynamics [17, 18, 36, 68, 73, 123, 126, 127, 131, 148]; adaptive cyber defense dynamics [25, 130]; proactive cyber defense dynamics [40]; and active cyber defense dynamics [132, 147].

1.4.1 Preventive and Reactive Cyber Defense Dynamics

The systematic preventive and reactive cyber defense dynamics model presented in [67] accommodates arbitrary, but time-independent, attack-defense structures $G = (V, E)$, push-based attacks (e.g., malware spreading), and pull-based attacks (e.g., drive-by download). The analytic result presented in [126] gives a sufficient condition (i.e., a specific parameter regime) under which the dynamics converge to a unique equilibrium, namely $\Pr(global\_state(t \to \infty) = 0) = 1$, meaning that all compromises will eventually be cleaned up. However, the properties of the dynamics in parameter regimes other than the specific regime characterized in [126] are not known until [148], which proves that the dynamics are *globally stable* in the *entire* parameter universe (i.e., the dynamics always converges to a unique equilibrium). This result remains true if the model parameters are extended to be node-dependent (i.e., different nodes $v \in V$ exhibit different cybersecurity characteristics, such as different host-based intrusion prevention/detection capabilities), and/or edge-dependent (i.e., different edges $e \in E$ exhibit different cybersecurity characteristics, such as different network-based intrusion prevention/detection capabilities) [148]. Moreover, the convergence speed is proven [148] to be exponential, except for a very special parameter regime (within which the dynamics converge polynomially). Although there is no closed-expression for the unique equilibrium, upper and lower bounds of the equilibrium can be obtained [126, 148]. Another important insight, which shows the value of theoretic studies, is that there is a practical statistical method that can be used to estimate the global cybersecurity state at equilibrium *without* knowing the model parameters, thanks to the global stability of the dynamics [126, 148].

The investigations mentioned above make the *independence* assumption that cyber attacks are waged independently of each other, which may not be the case when attacks are coordinated [118]. This highlights the importance of weakening, if not eliminating, the independence assumption. Initial results have been reported in [25, 123, 131]. An important finding is that assuming away the due dependence can lead to results that are unnecessarily restrictive, if not incorrect. Since the aforementioned dependence can be caused by multiple cyber attackers, preventive and reactive cyber defense dynamics have been extended to investigate the effect of multiple cyber attackers [127], which may even fight against each other. This leads to an interesting insight: the defender can leverage one attacker, say Alice, to "defeat" another attacker, say Bob, when the defender can more effectively defend against Alice than Bob.

In summary, we have a pretty deep understanding of preventive and reactive cyber defense dynamics. For example, the effectiveness of preventive and reactive cyber defenses is limited by a fundamental attack-defense asymmetry: the attack consequence is automatically amplified by a network effect reflected by the largest eigenvalue (in modulus) of the attack-defense structure G; in contrast, the defense effectiveness is not amplified by any network effect. This attack-defense asymmetry highlights the importance of enforcing strict network access control policy (e.g., direct communication between computers is allowed only when missions demand it), which effectively reduce the largest eigenvalue.

1.4.2 Adaptive Cyber Defense Dynamics

Cyber defense is often adaptive because the defender needs to adapt to the evolution of cyber attacks. Adaptive cyber defense dynamics have been investigated in [25, 130] while considering arbitrary attack-defense structure $G = (V, E)$. In [130], both semi-adaptive defenses (i.e., the defender dynamically adjusts the defense, but not necessarily geared towards the evolution of cyber attacks) and fully-adaptive defenses (i.e., the defender dynamically adjusts the defense geared towards the observed evolution of cyber attacks) are investigated. Adaptive control strategies can be used to force the dynamics to follow a trajectory that benefits the defender (e.g., forcing the dynamics to converge to a certain equilibrium). In [25], a new approach is proposed to model adaptive cyber defense dynamics with adaptive cyber attacks. An interesting finding is that the global cybersecurity state is relatively easy to quantify when the defense is either highly effectively or highly ineffective.

In summary, both cyber attacks and defenses are often adaptive, but they are challenging to model and analyze mathematically. For example, the intuitive concept of *adaptation agility* needs to be systematically investigated, with an initial effort presented in [79].

1.4.3 Proactive Cyber Defense Dynamics

Adaptive defenses may rely on the successful detection of attacks. Proactive defense does not suffer from this restriction because the defender can adjust the defense regardless of whether there are successful attacks or not. Moving-Target Defense (MTD) is a popular example of proactive defense. Many MTD techniques have been proposed (see, e.g., [90] and the numerous references therein) and many aspects of MTD have been investigated (see, e.g., [23, 51, 76, 83, 103]). However, very few studies have aimed at systematically quantifying the effectiveness of MTD. In what follows we outline a systematic use of MTD.

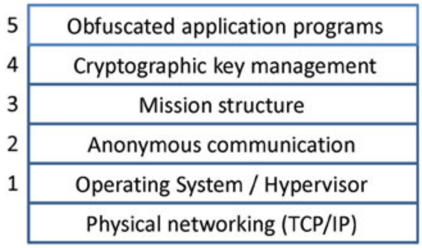

5 Obfuscated application programs
4 Cryptographic key management
3 Mission structure
2 Anonymous communication
1 Operating System / Hypervisor
Physical networking (TCP/IP)

Fig. 1.5 An example architecture showing that MTD can be employed at different layers, individually or collectively

As highlighted in Fig. 1.5, MTD can be employed at one or multiple layers of the software stack. Specifically, MTD can be employed at the operating system/hypervisor layer by frequently changing the underlying operating system/hypervisor environment (e.g., using VM migration). Anonymous communication can be leveraged to disrupt the attacker's reconnaissance capabilities by degrading the attacker's capability from waging *targeted and adaptive attacks* to *random attacks* [124, 125]. This means that anonymous communication can be leveraged for MTD to substantially increase the attacker's reconnaissance effort by dynamically adjusting, for example, the underlying anonymous communication infrastructure. At the *mission structure* layer, "mission structure" may be represented by a sub-graph $G_M(t) = (V_M(t), E_M(t))$ of the aforementioned attack-defense structure $G(t) = (V(t), E(t))$ with $V_M(t) \subseteq V(t)$ and $E_M(t) \subseteq E(t)$. In order to prevent the attacker from identifying a target node (e.g., the cyber command-and-control center), the defender can frequently relocate the target node. At the cryptographic key management layer, proactive cryptosystems [44], key-insulated cryptosystems [30–32], or leak-free cryptosystems [28, 29] can be used to tolerate the compromise of some computers, which hold some short-lived cryptographic key or cryptographic key shares [27, 124, 125]. Moreover, dynamic re-keying (e.g., [117, 149]) can be frequently enforced even in the absence of *detected* compromises because this can make the compromised cryptographic keys useless or can increase the chance that the compromise is detected [122]. At the application layer, the defender can use the following kinds of MTD to slow down the attacker: (1) re-obfuscating the application programs frequently; (2) dynamically re-shuffle honeypot IP addresses within a production network to capture new attacks [74].

While it is intuitive that MTD can be employed at each of these five layers, the main question is: When should the defender employ MTD and at which layers? Towards answering this question, the first systematic quantification study is presented in [40], which uses cybersecurity dynamics to quantify the effectiveness of MTD. However, the investigation treats MTD as a means, rather than a goal. That is, the effectiveness of MTD is *indirectly*, rather than directly, measured in [40]. In summary, proactive cyber defense is one of the very few approaches that can potentially defend against sophisticated attacks, such as zero-day attacks and Advanced Persistent Threats (APTs). More research needs to be done in order to systematically and directly quantify the effectiveness of proactive defense, including MTD.

1.4.4 Active Cyber Defense Dynamics

In the context of this chapter, active cyber defense means the use of "defenseware" (e.g., white worms or "malware killer" programs) to detect and clean up compromised computers. That is, active cyber defense is not about hacking back because it is employed within the administrative boundary of the network in question. The systematic modeling study of active cyber defense dynamics is initiated in [132], which formulates a mathematical model to quantify the effectiveness of active cyber defense. In active cyber defense dynamics, we need to consider a pair of attack-defense structures, denoted by $G_A(t) = (V_A(t), E_A(t))$ and $G_D(t) = (V_D(t), E_D(t))$. Note that $G_A(t)$ is centered at the attacker's point of view, and $G_D(t)$ is centered at the defender's point of view, while noting that it is possible that $G_A(t) = G_D(t)$. This leads to the identification of the optimal $G_D(t)$ under certain circumstances. In particular, it is shown [132] that active cyber defense can benefit the defender substantially by eliminating the aforementioned asymmetry, which is inherent to preventive and reactive cyber defense dynamics.

In [73], further investigation is conducted to identify optimal strategies for orchestrating active cyber defense against non-strategic or strategic attackers. In order to effectively defend against a non-strategic attacker, two flavors of optimal control strategies are investigated (i.e., *infinite-time horizon* control vs. *fast* control), by showing when the defender should adjust its active defense (including the extreme case of giving up the use of active defense, and instead using other kinds of defenses). In order to effectively defend against a strategic attacker, we identify Nash equilibrium strategies, while considering factors such as whether or not the attacker is willing to expose its advanced or zero-day attacks (exposure implying likelihood that these attacks will soon become useless).

In [147], it is shown for the first time that active cyber defense dynamics can exhibit *bifurcation* and *chaos*. Their cybersecurity implications include (1) it is not feasible or possible to seek to predict active cyber defense dynamics under certain circumstances, such as those reported in [147]; (2) the defender should seek to manipulate active cyber defense dynamics to avoid such "unmanageable" situations. In summary, the defender can use active cyber defense to offset the asymmetry advantage of the attacker in preventive and reactive cyber defense dynamics. However, active cyber defense is no panacea, and should be used together with other kinds of defenses [73]. Additional research needs to be conducted to deepen our understanding of active cyber defense dynamics.

1.5 Cybersecurity Data Analytics

Like cybersecurity first-principle modeling and analysis, cybersecurity data analytics is also centered at some well-defined cybersecurity metrics. However, cybersecurity data analytics is complementary to the cybersecurity first-principle modeling and analysis because the former is data- and experiment-driven (rather than assumption- or semantics-driven). More specifically, cybersecurity data analytics aim to achieve a range of objectives, including: (1) obtaining model parameters used by cybersecurity first-principle models, (2) validating or invalidating the assumptions made by cybersecurity first-principle models, and (3) helping tackle the *transient behavior* of cybersecurity dynamics. The state-of-the-art is that significant progress has been made in the aforementioned objectives (1) and (3), which are reviewed below, but not in (2) due to the lack of real-world datasets.

1.5.1 Obtaining Model Parameters

Measuring the Attack-Defense Structure *G(t)* In cybersecurity first-principle modeling and analysis, obtaining the attack-defense structure $G(t)$ is typically, and legitimately, treated as an orthogonal effort because it copes with a different aspect of the cybersecurity problem. As discussed above, researchers have recently started to investigate how to represent networks and computers at finer-grained granularities [17, 18]. Once a modeling resolution is determined, we need to represent the software stack (including applications and operating systems) of individual computers, represent individual computers as well as the dependence and communication relations within individual computers, represent the communication relations between computers (e.g., which computer or application is authorized to communicate with which other computer or application in a network), and represent the communication relation between a network and its external environment networks.

Measuring Susceptibility of Software Systems Cybersecurity first-principle models often assume parameters describing the *susceptibility* of an "atom" (e.g., computer or software component). In order to measure this parameter or metric, we need to measure the vulnerability of the "atom". From the perspective of software vulnerability, we need to measure to what extent a software program is vulnerable and susceptible to exploits. For this purpose, we need to understand and characterize the capabilities of *vulnerability detection* capabilities. For example, static analysis of software source code is one approach to detecting vulnerabilities. This approach can be further divided into two methods: *code similarity-based* [61, 69] vs. *pattern-based* [14, 34, 38, 70, 71, 85, 107, 136, 137]. The former method is effective in detecting vulnerabilities caused by certain kinds of code cloning [70]. Pattern-based methods are not limited to detecting clone-caused vulnerabilities. Pattern-based detection methods detect vulnerabilities at a coarse granularity, such as at the level of individual programs [38], individual components [85], individual files [111], or individual functions [135, 136]. More recent studies focus on fine-grained vulnerability detection [61, 69–71]. Studies in vulnerability detection represent a first step towards quantifying the susceptibility of software systems.

Measuring Defense Capabilities The accurate measurement of defense capabilities is one outstanding open problem. For example, most existing measurements often assume the availability of the ground truth in question. In the real world, ground truth is difficult to obtain. Therefore, it is important to investigate to what extent we can get rid of the ground truth, if possible at all. In the context of evaluating the detection capabilities of malware detectors, this problem has been investigated in [13, 33, 56]. In particular, statistical estimators are designed and evaluated in [33]. Moreover, relative accuracy of malware detectors, rather than absolute accuracy, can be estimated under much weaker assumptions [13].

Quantifying Attack Capabilities In order to measure the capabilities of pull-based attacks (e.g., drive-by download), it is necessary to measure the extent at which malicious websites can evade detection systems. There have been many proposals for detecting malicious websites (see, e.g., [75, 128]). However, the open problem is that the attacker, who knows the detection model or the dataset from which the model is learned, can manipulate the malicious websites to evade the detection systems in question. The investigation of this problem is initiated in [129], but there are no satisfactory solutions yet. For example, the proactive training approach used in [129] can only make the detection accuracy around 70–80%, which is far from sufficient.

1.5.2 Tackling the Transient Behavior Barrier

Towards ultimately tackling the transient behavior barrier, Fig. 1.6 highlights the "grey-box" statistical methodology initiated in [138]. The term "grey-box" means that the methodology first aims to characterize the statistical properties exhibited by the data (e.g., long-range dependence, extreme value, dependence, burstiness), and then uses these properties to guide the development of prediction models.

Progress in Coping with Cybersecurity Univariate Time Series A particular kind of univariate time series, dubbed *stochastic cyber attack processes*, has been substantially investigated [15, 95, 134, 138, 140]. These cyber attack processes describe the number of cyber attacks or incidents against a target of interest (e.g., a network, a computer, or even a particular port). Specifically, leveraging

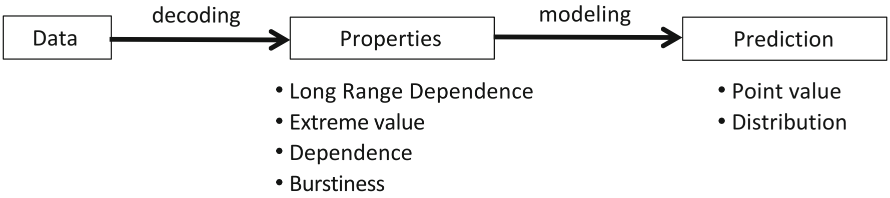


Fig. 1.6 The "grey-box" statistical methodology for cybersecurity data analytics

the statistical property known as *long-range dependence*, which is exhibited by the stochastic cyber attack processes corresponding to a dataset collected at a honeypot, the "grey-box" methodology leads to an 80% accuracy in forecasting the number of attacks coming to a network 1 h ahead of time. By further extending the model to accommodate the *extreme values* exhibited by the dataset, the 1-h ahead forecasting accuracy is improved to 88% [140]. A preliminary analysis of the spatiotemporal predictability shows that the forecasting upper bound is around 93% [15]. Focusing on the extreme values only, a *marked point process* model is developed to forecast the distribution of extreme values with good accuracy [95]. Another study focuses on the statistical analysis of breach incidents occurring between 2005 and 2017 [134], which shows that in contrast to previous beliefs, both the inter-arrival times and the breach sizes of hacking breach incidents should be described using stochastic processes, rather than probabilistic distributions, because of the autocorrelations exhibited by the data. These properties can be exploited to build accurate forecasting models [134]. These results evidently show predictability in cyberspace, at least from the perspectives that have been explored.

Progress in Coping with Cybersecurity Multivariate Time Series Many cybersecurity datasets can be represented by multivariate time series. The first investigation of this kind is to characterize and forecast the effectiveness of cyber defense early-warnings [133]. The idea of early-warning is to filter the cyber attacks, which are detected at cyber defense instruments (e.g., honeypot [101] or network telescope [10]) or third parties [74], against a network of interest. A unique research challenge, when compared with univariate time series, is to cope with the dependence between the time series, which manifests the dependence barrier [119] from a statistical perspective. For this purpose, the copula technique [53] turns out to be useful. A more general investigation of multivariate time series of cyber risks is conducted in [96]. The idea is to use a Copula-GARCH model to describe the multivariate dependence between stochastic cyber attack processes. In [96, 133], it is shown that assuming away the due dependence between stochastic cyber attack processes (i.e., the time series) can cause a severe underestimation of cybersecurity risks.

Progress in Coping with Cybersecurity Multivariate Time Series Many cybersecurity datasets can be represented by graph time series. A concrete example is the reconnaissance behaviors of cyber attackers, which can be represented as a time series of bipartite graphs [37, 139], which reflects one particular kind of the aforementioned attack-defense structure $G(t) = (V(t), E(t))$ over time t. For studying such time series of graphs, a systematic methodology is presented in [37]. At a high level, the methodology is to characterize the evolution (i.e., time series) of the *similarity* between two adjacent graphs $G(t)$ and $G(t+1)$, where the notion of *similarity* can have many different definitions (leading to various kinds of analyses). Using a real-world dataset, it is shown, among other things, that a couple of time resolutions are sufficient to accommodate and describe the *temporal* characteristics of these time series. This finding offers an effective guideline in coping with real-time data streams of this kind in real-world defense operations.

1.6 Future Research Directions

In this section we discuss future research directions with respect to the three axes mentioned above.

1.6.1 Cybersecurity Metrics

Towards ultimately tackling the problem cybersecurity metrics, the following two outstanding issues need to be resolved as soon as possible.

- Identifying a systematic, ideally complete, set of metrics that must be measured: Although many metrics have been proposed in the literature [94], the state-of-the-art is still that we do not know which metrics are essential to define and measure. This is because most existing metrics are introduced simply because they can be measured; in contrast, we need to know what metrics must be measured [98]. A fundamental question is: What kinds of metrics have to be measured in order to quantify cybersecurity? Therefore, we need to know a systematic set of metrics that can adequately describe cybersecurity. Better yet, it is important to know if there is a *complete* set of metrics, where "complete" means that any metric of interest can be derived from this set of metrics. Moreover, it is important to investigate the cost for measuring each of these metrics. This is because if one metric is costly to measure, we may need to seek easy-to-measure, alternate metric(s) to replace the hard-to-measure one as long as the former can answer the same kinds of questions as the latter does.
- Investigating mathematical properties of cybersecurity metrics and the operators that can be applied to them: As mentioned above, cybersecurity can be characterized at multiple model resolutions, reminiscent of the idea of considering security at multiple layers of abstractions [65]. Ideally, cybersecurity metrics at a lower model resolution (i.e., a higher level of abstraction or macroscopic cybersecurity) should be a mathematical function of the cybersecurity metrics defined and measured at some higher model resolutions (i.e., lower levels of abstractions or microscopic cybersecurity). This incurs the issue of aggregating lower levels of cybersecurity metrics into higher levels metrics [94, 98, 99]. For this purpose, we need to investigate the mathematical properties that should be satisfied by cybersecurity metrics, including axiomatic properties.

1.6.2 Cybersecurity First-Principle Modeling and Analysis

The following unique set of technical barriers need to be adequately tackled.

- The *scalability* barrier [119]: A first-principle, native approach to modeling the evolution of global cybersecurity state caused by the attack-defense interactions would be Stochastic Process models, which would incur an exponentially-large state space that is not tractable in general. How can this problem be tackled while preserving information in the model as much as possible? The current approach is to use mean-field style treatment, which essentially reduces the number of dimensions from exponentially-many to a number of dimensions that is proportional to the size of the attack-defense structure $G(t)$.

- The *nonlinearity* barrier [119]: It has been hypothesized that Cybersecurity Dynamics models are often highly nonlinear. The lack of real-world data has hindered the validation or rejection of this hypothesis, while many researchers believe in the nonlinearity. Coping with nonlinearity is a well known hard problem in general.
- The *dependence* barrier [119]: The security states of the "atoms" are not independent of each other because, for example, some software may have the same vulnerabilities. It is an outstanding open problem to cope with the dependence between the security state of the "atoms" (i.e., random variables), for which initial progress has been made as mentioned above [25, 131].
- The *structural dynamics* barrier [119]: The attack-defense structure $G(t)$ itself evolves over time. There have been some studies on accommodating specific kind of evolutions (see, for example, [40]). However, we need to establish a mathematical description of general evolution of $G(t)$.
- The *transient behavior* barrier [119]: Existing first-principle models often analyze the asymptotic behaviors of Cybersecurity Dynamics as time $t \rightarrow \infty$ (i.e., for sufficiently large t). For cybersecurity purposes, it is perhaps even more interesting to characterize the evolution of the global cybersecurity state before the dynamics converge to an equilibrium, if it does at all. This manifests the importance of cybersecurity data analytics. Despite the progress reviewed above (e.g., [15, 95, 134, 138, 140]), our understanding of the problem is still at the infant stage.
- The *uncertainty* barrier: Cybersecurity first-principle modeling often assumes the availability of complete information, meaning that the model parameters can be obtained precisely. This represents a first-step in building analytic models for baseline understanding. In practice, model parameters may not be known or may not be precisely measured, which highlights the importance of quantifying the consequences caused by uncertainties in the models and/or parameters.
- The *deception* barrier: In the cybersecurity domain, data or information not only can be missing or noisy, but also can be malicious because the attacker can intentionally inject or manipulate the measurements in question to mislead the defenders. This kind of deceptive data/information needs to be rigorously treated.
- The *human factor* barrier: The degrees that human users or defenders are vulnerable to social-engineering attacks need to be measured and quantified.

1.6.3 Cybersecurity Data Analytics

The following research problems need to be adequately resolved as soon as possible.

- Building a full-fledged statistical methodology to forecast holistic cybersecurity situational awareness, including the emergence of software zero-day vulnerabilities and attacks exploiting them. Although the studies reviewed above already showed the feasibility of predicting cybersecurity situational awareness from certain specific perspectives [15, 95, 96, 133, 134, 138–140], these results only represent a first step towards the ultimate goal.
- Tackling the dependence barrier as manifested in cybersecurity data analytics. Cybersecurity data can have extremely high dimensions, while dependence can be inherent to them. Therefore, we need to investigate forecasting models that can adequately accommodate the dependence "encoded" in real-world data. The results mentioned above [96, 133] only address a small tip of the iceberg.

1.7 Related Work

Prior Studies Related to the Cybersecurity Dynamics Foundation The present chapter systematically refines and extends an earlier treatment of Cybersecurity Dynamics [119], while accommodating the many advancements during the past few years. Although there have been investigations on exploring the various aspects (or characteristics) of the science of cybersecurity [42, 64, 106, 108, 112], to the best of our knowledge, we are the first to systematically map out a concrete framework as reviewed in the present chapter.

Prior Studies Related to Cybersecurity Metrics There are several recent surveys related to cybersecurity metrics [21, 89, 94, 104]. Moreover, the problem has been rejuvenated by new efforts [17, 18, 20, 21, 79, 89, 94, 104]. We treat cybersecurity metrics systematically, as highlighted in Eq. (1.1), for describing the configurations of networks, for describing systems and human vulnerabilities, for describing defense postures, for describing cyber attacks (i.e., threat models), and for describing the global cybersecurity state or cybersecurity situational awareness.

Prior Studies Related to Cybersecurity First-Principle Modeling and Analysis As discussed in [119], cybersecurity first-principle modeling is inspired by multiple endeavors in several disciplines, including: (1) Biological epidemic models [8, 9, 45, 60, 78]: These models have been adapted to the Internet setting (or cyber epidemic models) since Kephart and White [58, 59]. Later efforts aim to accommodate general network structures, including power-law network structures [11, 82, 86, 92, 92, 93] and *arbitrary* network structures (e.g., [12, 35, 115, 116]). (2) Interacting particle systems [72]: These models investigate the collective behaviors of interacting components and the phenomena that can emerge from these interactions. (3)

Microfoundation in Economics [47]: This effort aims to make connections between macroeconomic models to the underlying microeconomic models. However, the aforementioned technical barriers distinguish cybersecurity first-principle models from the models in the literature mentioned above. Moreover, it is the Cybersecurity Dynamics foundation that stresses that the attack-defense structure reflects, among other things, the access control policies that are enforced in a network, rather than the physical communication network. Furthermore, the foundation offers a unique set of cyber defense dynamics as reviewed above: preventive and reactive cyber defense dynamics, adaptive cyber defense dynamics, proactive cyber defense dynamics, and active cyber defense dynamics.

It is worth mentioning that cybersecurity first-principle models in the context of Cybersecurity Dynamics are different from the models in the context of Attack Graphs (see, for example, [2, 7, 19, 46, 52, 100, 105, 110]). This is because models in the context of Attack Graphs are *combinatorial* in nature (e.g., computing or enumerating attack paths with respect to a target); in contrast, models in the context of Cybersecurity Dynamics are *stochastic processes* in nature because they explicitly model the evolution of the global cybersecurity state over time t. This explains why these models are, as mentioned above, inspired by Biological Epidemic Models, Interacting Particle Systems, and Microfoundation in Economics, and why we can model various kinds of cyber defense dynamics.

Prior Studies Related to Cybersecurity Data Analytics There are numerous data-driven cybersecurity research activities, which however are often geared towards some specific events, attacks, or defenses. For example, honeypot-captured cyber attack data have been used for purposes of visualization [43], clustering attacks [4–6, 24], and characterizing attack behaviors such as inter-arrival times [1, 55]. In contrast, cybersecurity data analytics in the context of the present chapter is meant to become an inherent pillar of the Cybersecurity Dynamics foundation, by interacting with the other two pillars (i.e., cybersecurity first-principle modeling and analysis and cybersecurity metrics) as shown in Fig. 1.4.

1.8 Conclusion

We have systematically reviewed the Cybersecurity Dynamics foundation, with emphasis on the three active research axes or pillars in cybersecurity metrics, cybersecurity first-principle modeling and analysis, and cybersecurity data analytics. We discussed the progress in each of these axes and future research directions. We hope that we have clearly and successfully conveyed the following message: This is an exciting, but challenging, research endeavor that deserves a community wide effort to explore. We hope that the present chapter will inspire many more studies towards achieving the ultimate, full-fledged Cybersecurity Dynamics foundation for advancing the Science of Cybersecurity.

Acknowledgements This work was supported in part by ARO Grant # W911NF-17-1-0566 and ARL Grant # W911NF-17-2-0127. The author would like to thank his mentors for their encouragement, and his collaborators (including his former and current students) for deepening his understanding of the problem and potential solutions. The author would also like to thank Lisa Ho and John Charlton for proofreading the present chapter.

References

1. E. Alata, M. Dacier, Y. Deswarte, M. Kaaâniche, K. Kortchinsky, V. Nicomette, V. Pham, F. Pouget, Collection and analysis of attack data based on honeypots deployed on the internet, in *Proceedings of the Quality of Protection - Security Measurements and Metrics* (2006), pp. 79–91
2. M. Albanese, S. Jajodia, S. Noel, Time-efficient and cost-effective network hardening using attack graphs, in *Proceedings of the IEEE DSN'12* (2012), pp. 1–12
3. R. Albert, H. Jeong, A. Barabasi, Error and attack tolerance of complex networks. Nature **406**, 378–482 (2000)
4. S. Almotairi, A. Clark, M. Dacier, C. Leita, G. Mohay, V. Pham, O. Thonnard, J. Zimmermann, Extracting inter-arrival time based behaviour from honeypot traffic using cliques, in *5th Australian Digital Forensics Conference* (2007), pp. 79–87
5. S. Almotairi, A. Clark, G. Mohay, J. Zimmermann, Characterization of attackers' activities in honeypot traffic using principal component analysis, in *Proceedings of the IFIP International Conference on Network and Parallel Computing* (2008), pp. 147–154
6. S. Almotairi, A. Clark, G. Mohay, J. Zimmermann, A technique for detecting new attacks in low-interaction honeypot traffic, in *Proceedings of the International Conference on Internet Monitoring and Protection* (2009), pp. 7–13
7. P. Ammann, D. Wijesekera, S. Kaushik, Scalable, graph-based network vulnerability analysis, in *Proceedings of the ACM CCS'02* (2002), pp. 217–224
8. R. Anderson, R. May, *Infectious Diseases of Humans* (Oxford University Press, Oxford, 1991)
9. N. Bailey, *The Mathematical Theory of Infectious Diseases and Its Applications*, 2nd edn. (Griffin, London, 1975)
10. M. Bailey, E. Cooke, F. Jahanian, J. Nazario, D. Watson, Internet motion sensor: a distributed blackhole monitoring system, in *Proceedings of The 12th Network and Distributed System Security Symposium (NDSS'05)*, 2005
11. A. Barrat, M. Barthlemy, A. Vespignani, *Dynamical Processes on Complex Networks* (Cambridge University Press, Cambridge, 2008)
12. D. Chakrabarti, Y. Wang, C. Wang, J. Leskovec, C. Faloutsos, Epidemic thresholds in real networks. ACM Trans. Inf. Syst. Secur. **10**(4), 1–26 (2008)
13. J. Charlton, P. Du, J. Cho, S. Xu, Measuring relative accuracy of malware detectors in the absence of ground truth, in *Proceedings of IEEE MILCOM* (2018), pp. 450–455
14. Checkmarx (2018). https://www.checkmarx.com/
15. Y.-Z. Chen, Z.-G. Huang, S. Xu, Y.-C. Lai, Spatiotemporal patterns and predictability of cyberattacks. PLoS One **10**(5), e0124472 (2015)
16. H. Chen, D. Zou, S. Xu, H. Jin, B. Yuan, Y. Lu, SAND: semi-automated adaptive network defense via programmable rule generation and deployment (2019, manuscript under review)
17. H. Chen, J.-H. Cho, S. Xu, Quantifying the security effectiveness of firewalls and DMZs, in *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (HoTSoS'2018)* (2018), pp. 9:1–9:11
18. H. Chen, J.-H. Cho, S. Xu, Quantifying the security effectiveness of network diversity: poster, in *Proceedings of the 5th Annual Symposium and Bootcamp on Hot Topics in the Science of Security (HoTSoS'2018)* (2018), p. 24:1

19. Y. Cheng, J. Deng, J. Li, S. DeLoach, A. Singhal, X. Ou, Metrics of security, in *Cyber Defense and Situational Awareness*, vol. 62 (Springer, Cham, 2014)
20. J.-H. Cho, P. Hurley, S. Xu, Metrics and measurement of trustworthy systems, in *IEEE Military Communication Conference (MILCOM 2016)*, 2016
21. J. Cho, S. Xu, P. Hurley, M. Mackay, T. Benjamin, M. Beaumont, STRAM: measuring the trustworthiness of computerbased systems, ACM Computing Survey, Accepted for publication (to appear in 2019)
22. J. Chow, B. Pfaff, T. Garfinkel, K. Christopher, M. Rosenblum, Understanding data lifetime via whole system simulation, in *Proceedings of Usenix Security Symposium 2004*, 2004
23. W. Connell, D.A. Menascé, M. Albanese, Performance modeling of moving target defenses, in *Proceedings of the 2017 Workshop on Moving Target Defense*, MTD '17 (2017), pp. 53–63
24. G. Conti, K. Abdullah, Passive visual fingerprinting of network attack tools, in *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security* (2004), pp. 45–54
25. G. Da, M. Xu, S. Xu, A new approach to modeling and analyzing security of networked systems, in *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)* (2014), pp. 6:1–6:12
26. D. Dagon, G. Gu, C.P. Lee, W. Lee, A taxonomy of botnet structures, in *23rd Annual Computer Security Applications Conference (ACSAC'07)* (2007), pp. 325–339
27. Y. Desmedt, Y. Frankel, Threshold cryptosystems, in *Proceedings of the CRYPTO 89* (1989), pp. 307–315
28. X. Ding, G. Tsudik, S. Xu, Leak-free group signatures with immediate revocation, in *24th International Conference on Distributed Computing Systems (ICDCS 2004)* (IEEE Computer Society, Los Alamitos, 2004), pp. 608–615
29. X. Ding, G. Tsudik, S. Xu, Leak-free mediated group signatures. J. Comput. Secur. **17**(4), 489–514 (2009)
30. Y. Dodis, J. Katz, S. Xu, M. Yung, Key-insulated public key cryptosystems, in *Advances in Cryptology - EUROCRYPT 2002*, ed. by L.R. Knudsen. Lecture Notes in Computer Science, vol. 2332 (Springer, Berlin, 2002), pp. 65–82
31. Y. Dodis, J. Katz, S. Xu, M. Yung, Strong key-insulated signature schemes, in *Public Key Cryptography (PKC'03)* (2003), pp. 130–144
32. Y. Dodis, W. Luo, S. Xu, M. Yung, Key-insulated symmetric key cryptography and mitigating attacks against cryptographic cloud software, in *7th ACM Symposium on Information, Computer and Communications Security, ASIACCS '12* (2012), pp. 57–58
33. P. Du, Z. Sun, H. Chen, J.H. Cho, S. Xu, Statistical estimation of malware detection metrics in the absence of ground truth. IEEE Trans. Inf. Forensics Secur. **13**, 2965–2980 (2018)
34. Flawfinder (2018). http://www.dwheeler.com/flawfinder
35. A. Ganesh, L. Massoulie, D. Towsley, The effect of network topology on the spread of epidemics, in *Proceedings of IEEE Infocom 2005* (2005)
36. R. Garcia-Lebron, D.J. Myers, S. Xu, J. Sun, Node diversification in complex networks by decentralized colouring. J. Complex Networks, cny031. (2018). https://doi.org/10.1093/comnet/cny031
37. R. Garcia-Lebron, K. Schweitzer, R. Bateman, S. Xu, A framework for characterizing the evolution of cyber attackervictim relation graphs, in *Proceedings of IEEE MILCOM* (2018), pp. 70–75
38. G. Grieco, G.L. Grinblat, L.C. Uzal, S. Rawat, J. Feist, L. Mounier, Toward large-scale vulnerability discovery using machine learning, in *Proceedings of the Sixth ACM on Conference on Data and Application Security and Privacy, CODASPY 2016*, New Orleans (2016), pp. 85–96
39. L. Guan, J. Lin, B. Luo, J. Jing, J. Wang, Protecting private keys against memory disclosure attacks using hardware transactional memory, in *Proceedings of the 2015 IEEE Symposium on Security and Privacy, SP '15* (2015), pp. 3–19
40. Y. Han, W. Lu, S. Xu, Characterizing the power of moving target defense via cyber epidemic dynamics, in *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)* (2014), pp. 10:1–10:12

41. K. Harrison, S. Xu, Protecting cryptographic keys from memory disclosures, in *Proceedings of the 2007 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN-DCCS'07)* (IEEE Computer Society, Los Alamitos, 2007), pp. 137–143
42. C. Herley, P.C.v. Oorschot, SoK: science, security and the elusive goal of security as a scientific pursuit, in *2017 IEEE Symposium on Security and Privacy (SP)*, May 2017, pp. 99–120
43. A. Herrero, U. Zurutuza, E. Corchado, A neural-visualization IDS for honeynet data. Int. J. Neural Syst. **22**(2), 1250005 (2012)
44. A. Herzberg, M. Jakobsson, S. Jarecki, H. Krawczyk, M. Yung, Proactive public key and signature schemes, in *Proceedings of the Fourth Annual Conference on Computer and Communications Security* (ACM, New York, 1997), pp. 100–110
45. H. Hethcote, The mathematics of infectious diseases. SIAM Rev. **42**(4), 599–653 (2000)
46. J. Homer, S. Zhang, X. Ou, D. Schmidt, Y. Du, S. Raj Rajagopalan, A. Singhal, Aggregating vulnerability metrics in enterprise networks using attack graphs. J. Comput. Secur. **21**(4), 561–597 (2013)
47. K. Hoover, Idealizing reduction: the microfoundations of macroeconomics. Erkenntnis **73**, 329–347 (2010)
48. A. Hussain, J. Heidemann, C. Papadopoulos, A framework for classifying denial of service attacks, in *Proceedings of ACM SIGCOMM'03* (2003), pp. 99–110
49. E.M. Hutchins, M.J. Cloppert, R.M. Amin, Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains, in *2011 International Conference on Information Warfare and Security* (2011)
50. INFOSEC Research Council, Hard problem list (2007). http://www.infosec-research.org/docs_public/20051130-IRC-HPL-FINAL.pdf
51. J.H. Jafarian, E. Al-Shaer, Q. Duan, Openflow random host mutation: transparent moving target defense using software defined networking, in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN'12)* (2012), pp. 127–132
52. S. Jha, O. Sheyner, J. Wing, Two formal analyses of attack graphs, in *Proceedings of the IEEE Workshop on Computer Security Foundations* (2002), pp. 49–59
53. H. Joe, *Dependence Modeling with Copulas* (CRC Press, Boca Raton, 2014)
54. A. Juels, B.S. Kaliski Jr., Pors: proofs of retrievability for large files, in *Proceedings of the ACM Conference on Computer and Communications Security (CCS'07)* (2007), pp. 584–597
55. M. Kaâniche, Y. Deswarte, E. Alata, M. Dacier, V. Nicomette, Empirical analysis and statistical modeling of attack processes based on honeypots. CoRR (2007). http://arxiv.org/abs/0704.0861
56. A. Kantchelian, M.C. Tschantz, S. Afroz, B. Miller, V. Shankar, R. Bachwani, A.D. Joseph, J.D. Tygar, Better malware ground truth: techniques for weighting anti-virus vendor labels, in *Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security* (ACM, New York, 2015), pp. 45–56
57. E.J. Kartaltepe, J.A. Morales, S. Xu, R.S. Sandhu, Social network-based botnet command-and-control: emerging threats and countermeasures, in *ACNS* (2010), pp. 511–528
58. J. Kephart S. White, Directed-graph epidemiological models of computer viruses, in *IEEE Symposium on Security and Privacy* (1991), pp. 343–361
59. J. Kephart, S. White, Measuring and modeling computer virus prevalence, in *IEEE Symposium on Security and Privacy* (1993), pp. 2–15
60. W. Kermack, A. McKendrick, A contribution to the mathematical theory of epidemics. Proc. R. Soc. Lond. A **115**, 700–721 (1927)
61. S. Kim, S. Woo, H. Lee, H. Oh, VUDDY: a scalable approach for vulnerable code clone discovery, in *2017 IEEE Symposium on Security and Privacy* (2017), pp. 595–614
62. P. Kocher, D. Genkin, D. Gruss, W. Haas, M. Hamburg, M. Lipp, S. Mangard, T. Prescher, M. Schwarz, Y. Yarom, Spectre attacks: exploiting speculative execution. CoRR (2018). http://arxiv.org/abs/1801.01203
63. B. Köpf, D. Basin, An information-theoretic model for adaptive side-channel attacks, in *Proceedings of the ACM Conference on Computer and Communications Security* (ACM, New York, 2007), pp. 286–296

64. A. Kott, Towards fundamental science of cyber security, in *Network Science and Cybersecurity*, ed. by R.E. Pino. Advances in Information Security, vol. 55 (Springer, New York, 2014), pp. 1–13
65. B. Lampson, Practical principles for computer security (2006). http://bwlampson.site/Slides/PracticalPrinciplesSecurityAbstract.htm
66. J. Leonard, S. Xu, R.S. Sandhu, A framework for understanding botnets, in *Proceedings of the Fourth International Conference on Availability, Reliability and Security, ARES* (2009), pp. 917–922
67. X. Li, P. Parker, S. Xu, Towards quantifying the (in)security of networked systems, in *21st IEEE International Conference on Advanced Information Networking and Applications (AINA'07)* (2007), pp. 420–427
68. X. Li, P. Parker, S. Xu, A stochastic model for quantitative security analysis of networked systems. IEEE Trans. Dependable Secure Comput. **8**(1), 28–43 (2011)
69. Z. Li, D. Zou, S. Xu, H. Jin, H. Qi, J. Hu, VulPecker: an automated vulnerability detection system based on code similarity analysis, in *Proceedings of the 32nd Annual Conference on Computer Security Applications, ACSAC*, Los Angeles (2016), pp. 201–213
70. Z. Li, D. Zou, S. Xu, X. Ou, H. Jin, S. Wang, Z. Deng, Y. Zhong, VulDeePecker: a deep learning-based system for vulnerability detection, in *Proceedings of the 25th Annual Network and Distributed System Security Symposium (NDSS'2018)* (2018)
71. Z. Li, D. Zou, S. Xu, H. Jin, Y. Zhu, Z. Chen, S. Wang, J. Wang, SySeVR: a framework for using deep learning to detect software vulnerabilities, CoRR abs/1807.06756 (2018)
72. T. Liggett, *Interacting Particle Systems* (Springer, Berlin, 1985)
73. W. Lu, S. Xu, X. Yi, Optimizing active cyber defense dynamics, in *Proceedings of the 4th International Conference on Decision and Game Theory for Security (GameSec'13)* (2013), pp. 206–225
74. W. Luo, L. Xu, Z. Zhan, Q. Zheng, S. Xu, Federated cloud security architecture for secure and agile clouds, in *High Performance Cloud Auditing and Applications*, ed. by K.J. Han, B.-Y. Choi, S. Song (Springer, New York, 2014), pp. 169–188
75. J. Ma, L.K. Saul, S. Savage, G.M. Voelker, Learning to detect malicious urls. ACM TIST **2**(3), 30:1–30:24 (2011)
76. H. Maleki, S. Valizadeh, W. Koch, A. Bestavros, M. van Dijk, Markov modeling of moving target defense games, in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*, MTD '16 (2016), pp. 81–92
77. Mandiant. Apt1 report. https://www.fireeye.com/content/dam/fireeyewww/services/pdfs/mandiant-apt1-report.pdf, 16 Feb 2013. Accessed 08 July 2016
78. A. McKendrick, Applications of mathematics to medical problems. Proc. Edinb. Math. Soc. **14**, 98–130 (1926)
79. J. Mireles, E. Ficke, J.-H. Cho, P. Hurley, S. Xu, Metrics towards measuring cyber agility (2019, manuscript in submission)
80. A. Mohaisen, O. Alrawi, AV-meter: an evaluation of antivirus scans and labels, in *Detection of Intrusions and Malware, and Vulnerability Assessment - 11th International Conference, DIMVA 2014, Proceedings* (2014), pp. 112–131
81. J. Morales, S. Xu, R. Sandhu, Analyzing malware detection efficiency with multiple anti-malware programs, in *Proceedings of 2012 ASE International Conference on Cyber Security (CyberSecurity'12)* (2012)
82. Y. Moreno, R. Pastor-Satorras, A. Vespignani, Epidemic outbreaks in complex heterogeneous networks. Eur. Phys. J. B **26**, 521–529 (2002)
83. D. Mulamba, I. Ray, Resilient reference monitor for distributed access control via moving target defense, in *Data and Applications Security and Privacy XXXI*, ed. by G. Livraga, S. Zhu (2017), pp. 20–40
84. National Science and Technology Council, Trustworthy cyberspace: strategic plan for the federal cybersecurity research and development program (2011). https://www.nitrd.gov/SUBCOMMITTEE/csia/Fed_Cybersecurity_RD_Strategic_Plan_2011.pdf

85. S. Neuhaus, T. Zimmermann, C. Holler, A. Zeller, Predicting vulnerable software components, in *Proceedings of the 2007 ACM Conference on Computer and Communications Security, CCS 2007*, Alexandria (2007), pp. 529–540
86. M. Newman, The structure and function of complex networks. SIAM Rev. **45**, 167 (2003)
87. D.M. Nicol, W.H. Sanders, K.S. Trivedi, Model-based evaluation: from dependability to security. IEEE Trans. Dependable Secure Comput. **1**(1), 48–65 (2004)
88. D. Nicol, B. Sanders, J. Katz, B. Scherlis, T. Dumitra, L. Williams, M.P. Singh, The science of security 5 hard problems (Aug 2015). http://cps-vo.org/node/21590
89. S. Noel, S. Jajodia, *A Suite of Metrics for Network Attack Graph Analytics* (Springer, Cham, 2017), pp. 141–176
90. H. Okhravi, M. Rabe, T. Mayberry, W. Leonard, T. Hobson, D. Bigelow, W. Streilein, Survey of cyber moving targets (MIT Lincoln Lab technical report), 2013
91. T.P. Parker, S. Xu, A method for safekeeping cryptographic keys from memory disclosure attacks, in *First International Conference on Trusted Systems (INTRUST'2009)* (2009), pp. 39–59
92. R. Pastor-Satorras, A. Vespignani, Epidemic dynamics and endemic states in complex networks. Phys. Rev. E **63**, 066117 (2001)
93. R. Pastor-Satorras, A. Vespignani, Epidemic dynamics in finite size scale-free networks. Phys. Rev. E **65**, 035108 (2002)
94. M. Pendleton, R. Garcia-Lebron, J.-H. Cho, S. Xu, A survey on systems security metrics. ACM Comput. Surv. **49**(4), 62:1–62:35 (2016)
95. C. Peng, M. Xu, S. Xu, T. Hu, Modeling and predicting extreme cyber attack rates via marked point processes. J. Appl. Stat. **44**(14), 2534–2563 (2017)
96. C. Peng, M. Xu, S. Xu, T. Hu, Modeling multivariate cybersecurity risks. J. Appl. Stat **45**(15), 2718–2740 (2018). https://doi.org/10.1080/02664763.2018.1436701
97. R. Perdisci, U. ManChon, VAMO: towards a fully automated malware clustering validity analysis, in *Proceedings of the 28th Annual Computer Security Applications Conference*, ACSAC '12 (2012), pp. 329–338
98. S.L. Pfleeger, Useful cybersecurity metrics. IT Prof. **11**(3), 38–45 (2009)
99. S.L. Pfleeger, R.K. Cunningham, Why measuring security is hard. IEEE Secur. Priv. **8**(4), 46–54 (2010)
100. C. Phillips, L.P. Swiler, A graph-based system for network-vulnerability analysis, in *Proceedings of the 1998 Workshop on New Security Paradigms*, NSPW '98 (1998), pp. 71–79
101. N. Provos, A virtual honeypot framework, in *USENIX Security Symposium* (2004), pp. 1–14
102. N. Provos, D. McNamee, P. Mavrommatis, K. Wang, N. Modadugu, The ghost in the browser analysis of web-based malware, in *Proceedings of the First Workshop on Hot Topics in Understanding Botnets (HotBots'07)* (2007)
103. M.A. Rahman, E. Al-Shaer, R.B. Bobba, Moving target defense for hardening the security of the power system state estimation, in *Proceedings of the First ACM Workshop on Moving Target Defense, MTD '14* (2014), pp. 59–68
104. A. Ramos, M. Lazar, R.H. Filho, J.J.P.C. Rodrigues, Model-based quantitative network security metrics: a survey. IEEE Commun. Surv. Tutorials **19**(4), 2704–2734 (2017)
105. R.W. Ritchey, P. Ammann, Using model checking to analyze network vulnerabilities, in *Proceedings of the IEEE Symposium on Security and Privacy* (2000), pp. 156–165
106. A. Roque, K.B. Bush, C. Degni, Security is about control: insights from cybernetics, in *Proceedings of the Symposium and Bootcamp on the Science of Security*, Pittsburgh, April 19–21, 2016, pp. 17–24
107. Rough Audit Tool for Security (2014). https://code.google.com/archive/p/rough-auditing-tool-for-security/
108. F. Schneider, Blueprint for a science of cybersecurity. Technical report, Cornell University, May 2011. Also to appear in The Next Wave
109. Y. Shang, W. Luo, S. Xu, *l*-hop percolation on networks with arbitrary degree distributions and its applications. Phys. Rev. E **84**, 031113 (2011)
110. O. Sheyner, J. Haines, S. Jha, R. Lippmann, J. Wing, Automated generation and analysis of attack graphs, in *IEEE Symposium on Security and Privacy* (2002), pp. 273–284

111. Y. Shin, A. Meneely, L. Williams, J.A. Osborne, Evaluating complexity, code churn, and developer activity metrics as indicators of software vulnerabilities. IEEE Trans. Softw. Eng. **37**(6), 772–787 (2011)
112. J.M. Spring, T. Moore, D.J. Pym, Practicing a science of security: a philosophy of science perspective, in *Proceedings of the 2017 New Security Paradigms Workshop, NSPW 2017* (2017), pp. 1–18
113. C. Trippel, D. Lustig, M. Martonosi, Meltdownprime and spectreprime: automatically-synthesized attacks exploiting invalidation-based coherence protocols. CoRR (2018). http://arxiv.org/abs/1802.03802
114. A. Tyra, J. Li, Y. Shang, S. Jiang, Y. Zhao, S. Xu, Robustness of non-interdependent and interdependent networks against dependent and adaptive attacks. Phys. A Stat. Mech. Appl. **482**, 713–727 (2017)
115. P. Van Mieghem, J. Omic, R. Kooij, Virus spread in networks. IEEE/ACM Trans. Netw. **17**(1), 1–14 (2009)
116. Y. Wang, D. Chakrabarti, C. Wang, C. Faloutsos, Epidemic spreading in real networks: an eigenvalue viewpoint, in *Proceedings of the 22nd IEEE Symposium on Reliable Distributed Systems (SRDS'03)* (2003), pp. 25–34
117. S. Xu, On the security of group communication schemes. J. Comput. Secur. **15**(1), 129–169 (2007)
118. S. Xu, Collaborative attack vs. collaborative defense, in *4th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom'2008)* (2008), pp. 217–228
119. S. Xu, Cybersecurity dynamics, in *Proceedings of the Symposium and Bootcamp on the Science of Security (HotSoS'14)* (2014), pp. 14:1–14:2
120. S. Xu, Emergent behavior in cybersecurity, in *Proceedings of the 2014 Symposium and Bootcamp on the Science of Security (HotSoS'14)* (2014), pp. 13:1–13:2
121. S. Xu, Cybersecurity dynamics publications. http://www.cs.utsa.edu/~shxu/socs/
122. S. Xu, M. Yung, Expecting the unexpected: towards robust credential infrastructure, in *13th International Conference on Financial Cryptography and Data Security (FC'09)* (2009), pp. 201–221
123. M. Xu, S. Xu, An extended stochastic model for quantitative security analysis of networked systems. Internet Math. **8**(3), 288–320 (2012)
124. S. Xu, X. Li, P. Parker, Exploiting social networks for threshold signing: attack-resilience vs. availability, in *ACM Symposium on Information, Computer and Communications Security (ASIACCS'08)* (2008), pp. 325–336
125. S. Xu, X. Li, T. Parker, X. Wang, Exploiting trust-based social networks for distributed protection of sensitive data. IEEE Trans. Inf. Forensics Secur. **6**(1), 39–52 (2011)
126. S. Xu, W. Lu, L. Xu, Push- and pull-based epidemic spreading in arbitrary networks: thresholds and deeper insights. ACM Trans. Auton. Adapt. Syst. **7**(3), 32:1–32:26 (2012)
127. S. Xu, W. Lu, Z. Zhan, A stochastic model of multivirus dynamics. IEEE Trans. Dependable Secure Comput. **9**(1), 30–45 (2012)
128. L. Xu, Z. Zhan, S. Xu, K. Ye, Cross-layer detection of malicious websites, in *Third ACM Conference on Data and Application Security and Privacy (ACM CODASPY'13)* (2013), pp. 141–152
129. L. Xu, Z. Zhan, S. Xu, K. Ye, An evasion and counter-evasion study in malicious websites detection, in *IEEE Conference on Communications and Network Security (CNS'14)* (2013), pp. 141–152
130. S. Xu, W. Lu, L. Xu, Z. Zhan, Adaptive epidemic dynamics in networks: thresholds and control. ACM Trans. Auton. Adapt. Syst. **8**(4), 19 (2014)
131. M. Xu, G. Da, S. Xu, Cyber epidemic models with dependences. Internet Math. **11**(1), 62–92 (2015)
132. S. Xu, W. Lu, H. Li, A stochastic model of active cyber defense dynamics. Internet Math. **11**(1), 23–61 (2015)

133. M. Xu, L. Hua, S. Xu, A vine copula model for predicting the effectiveness of cyber defense early-warning. Technometrics **59**(4), 508–520 (2017)
134. M. Xu, K.M. Schweitzer, R.M. Bateman, S. Xu, Modeling and predicting cyber hacking breaches. IEEE Trans. Inf. Forensics Secur. **13**(11), 2856–2871 (2018)
135. F. Yamaguchi, F. "FX" Lindner, K. Rieck, Vulnerability extrapolation: assisted discovery of vulnerabilities using machine learning, in *Proceedings of the 5th USENIX Workshop on Offensive Technologies, WOOT'11*, 8 Aug 2011, San Francisco (2011), pp. 118–127
136. F. Yamaguchi, M. Lottmann, K. Rieck, Generalized vulnerability extrapolation using abstract syntax trees, in *28th Annual Computer Security Applications Conference, ACSAC 2012*, Orlando (2012), pp. 359–368
137. F. Yamaguchi, C. Wressnegger, H. Gascon, K. Rieck, Chucky: exposing missing checks in source code for vulnerability discovery, in *2013 ACM SIGSAC Conference on Computer and Communications Security, CCS'13*, Berlin (2013), pp. 499–510
138. Z. Zhan, M. Xu, S. Xu, Characterizing honeypot-captured cyber attacks: statistical framework and case study. IEEE Trans. Inf. Forensics Secur. **8**(11), 1775–1789 (2013)
139. Z. Zhan, M. Xu, S. Xu, A characterization of cybersecurity posture from network telescope data, in *Proceedings of the 6th International Conference on Trustworthy Systems (InTrust'14)* (2014), pp. 105–126
140. Z. Zhan, M. Xu, S. Xu, Predicting cyber attack rates with extreme values. IEEE Trans. Inf. Forensics Secur. **10**(8), 1666–1677 (2015)
141. Y. Zhao, Y. Xie, F. Yu, Q. Ke, Y. Yu, Y. Chen, E. Gillum, BotGraph: large scale spamming botnet detection, in *Proc. NSDI'09* (2009), pp. 321–334
142. Q. Zheng, S. Xu, Fair and dynamic proofs of retrievability, in *First ACM Conference on Data and Application Security and Privacy, (CODASPY'2011)* (2011), pp. 237–248
143. Q. Zheng, S. Xu, Secure and efficient proof of storage with deduplication, in *Second ACM Conference on Data and Application Security and Privacy (CODASPY'2012)* (2012), pp. 1–12
144. Q. Zheng, S. Xu, Verifiable delegated set intersection operations on outsourced encrypted data, in *2015 IEEE International Conference on Cloud Engineering, IC2E 2015* (2015), pp. 175–184
145. Q. Zheng, S. Xu, G. Ateniese, Efficient query integrity for outsourced dynamic databases, in *Proceedings of the 2012 ACM Workshop on Cloud Computing Security, CCSW 2012*, Raleigh, 19 Oct 2012, pp. 71–82
146. Q. Zheng, S. Xu, G. Ateniese, VABKS: verifiable attribute-based keyword search over outsourced encrypted data, in *Proceedings of the 2014 IEEE Conference on Computer Communications (INFOCOM'2014)* (2014), pp. 522–530
147. R. Zheng, W. Lu, S. Xu, Active cyber defense dynamics exhibiting rich phenomena, in *Proceedings of the 2015 Symposium and Bootcamp on the Science of Security (HotSoS'15)* (2015), pp. 2:1–2:12
148. R. Zheng, W. Lu, S. Xu, Preventive and reactive cyber defense dynamics is globally stable. IEEE Trans. Netw. Sci. Eng. **5**(2), 156–170 (2017)
149. S. Zhu, S. Setia, S. Xu, S. Jajodia, GKMPAN: an efficient group rekeying scheme for secure multicast in ad-hoc networks. J. Comput. Secur. **14**(4), 301–325 (2006)

Chapter 2
Proactive Network Defense with Game Theory

Sinong Wang and Ness Shroff

Abstract Traditional proactive network defenses deploy security resources in the network based on probabilistic policies to confuse potential attackers. However, this strategy can be exploited by stealthy attackers, leading to reduced efficiency and higher vulnerability. Game theory has been shown to provide a sound mathematical approach to overcome these deficiencies and determine an optimal defense strategy. However, existing game theoretic models typically either assume additive utility functions, or that the attacker can attack only one target. While such assumptions lead to tractable analyses, they miss key inherent dependencies that exist among different targets in current complex networks. In this chapter, we generalize the traditional security game model to the network scenario. We examine such a general security game from a theoretical perspective and provide a unified theoretical framework. In particular, we show that each security game is equivalent to a combinatorial optimization problem over a set system, which consists of defender's pure strategy space. The key technique we use is based on projection of a polytope based transformation, and the ellipsoid method. We also provide several important applications of our developed framework, and show that for several problem classes, optimal defense strategies can be developed in polynomial time. Our approach paves the way for a deeper investigation into using game theoretic techniques for solving designing security mechanisms in networks, and we conclude by outlining a number of important future directions that need to be investigated.

2.1 Introduction

Most critical systems use some type of proactive defense through firewalls, reinforcing systems through regular software updates, providing police protection of important locations, etc. However, one of the key problems in proactive network

S. Wang · N. Shroff (✉)
The Ohio State University, Columbus, OH, USA
e-mail: wang.7691@osu.edu; shroff.11@osu.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances
in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_2

defense is *how to efficiently allocate limited resources to protect targets in a network against potential threats*. For example, the government may have a limited police force to operate checkpoints and conduct random patrols over some city blocks, or have a limited number of coders that restricts how often and for what functionality new software updates are generated. However, the adversarial aspect in security domain poses a unique challenge for allocating resources. An intelligent attacker can observe the defender's strategy and gather information to schedule more effective attacks. Therefore, the simple random strategy of "rolling the dice" may be exploited by the attacker, which greatly reduces the effectiveness of the strategy. This is where game theory can help devise strategies that are optimal even under intelligent and stealthy attackers.

2.1.1 Why Game Theory?

Before we describe the importance of applying game theory to the proactive network defense, let us first look at the following example.

Example 2.1 As shown in Fig. 2.1, there exists a network with multiple nodes and links. The goal of defender or infrastructure service provider is to transmit the packets from the node *s* to node *d* along different paths. In practice, there might exists some hackers attempting to intercept the packet and subtract the confidential contents. To avoid interception from attackers, the defender can probabilistically choose a different routing path. For example, in the above network, we have four routing paths that has the possibility to confuse the attacker. However, the question is *is probabilistically mixing the strategy a secured policy in proactive network defense?* In practice, the stealthy attacker can observe the defender's probabilistic strategy and predict the defender's next move, which may lead to disastrous consequences.

With the development of computational game theory, such resource allocation problems can be cast in game-theoretic contexts, which provides a sounder math-

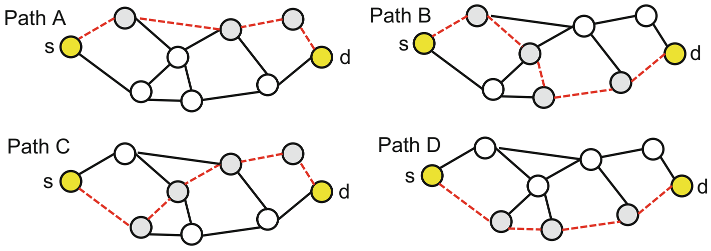


Fig. 2.1 A network with four possible routing paths. The yellow nodes are source and destination nodes. The grey nodes are intermediate nodes in the routing path

ematical approach to determine the optimal defense strategy. It allows the analyst to factor differential risks and values into the model, incorporate game-theoretic predictions of how the attacker would respond to the security policy, and finally determine an equilibrium strategy that cannot be exploited by adversaries to obtain a higher payoff. In the past decade, there has been an explosion of research attempting to address this approach, which has led to the development of well-known models of security games.

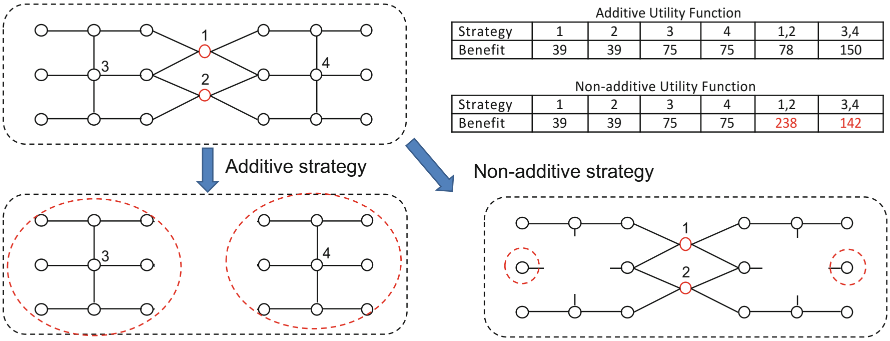
Moreover, it has become increasingly apparent that security failures in network and information systems are often caused by a misunderstanding of the incentives of the entities involved in the system instead of a lack of proper technical mechanisms [1, 2]. To this end, there exists game theoretical models trying to understanding this phenomenon using analytical approaches [3–6]. Some other recent works [7–9] also consider Advanced Persistent Threats (APT) in cyber security. APT attacks have several distinguishing properties that render traditional defense mechanism less effective. First, they are often launched by incentive driven entities with specific targets. Second, they are persistent in achieving the goals, and may involve multiple stages or continuous operations over a long period of time. Third, they are highly adaptive and stealthy, which requires the game model capturing the persistent and stealthy behavior of advanced attacks.

The classic *security game* is a two-player game played between a *defender* and an *attacker*. The attacker chooses one target to attack; The defender allocates (randomly) limited resources, subject to various domain constraints, to protect a set of targets. The attacker (defender) will obtain the benefits (losses) for those successfully attacked targets and losses (benefits) for those defended targets. The goal of the defender is to choose a random strategy so as to play optimally under some solution concepts such as Nash equilibrium and strong Stackelberg equilibrium. This *security game* model and its game-theoretic solution is *currently being used by many security agencies including US Coast Guard and Federal Air Marshals Service*(FAMS) [10], Transportation System Administration [11] and even in the wildlife protection [12]; see book by Tambe [13] for an overview.

2.1.2 Challenges in the Classical Security Game Model

Before we discuss the challenges in the classical security game model, let us first consider the following example.

Example 2.2 As shown in Fig. 2.2, we have a 20-node network. It is clear that nodes 1, 2, 3 and 4 are the critical battlefields in this network. Suppose that the attacker's and defender's strategies are {1}, {2}, {3}, {1, 2} or {3, 4}, where $\{v\}$ denotes the index of the nodes. We adopt the network value proposed by Gueye et al. [14] as the security measure for different nodes, which calculates the importance of a group of nodes by subtracting the value of the network by removing these nodes from



Additive Utility Function

Strategy	1	2	3	4	1,2	3,4
Benefit	39	39	75	75	78	150

Non-additive Utility Function

Strategy	1	2	3	4	1,2	3,4
Benefit	39	39	75	75	238	142

Fig. 2.2 Example of security game in a 20-nodes network with independent targets assumption (additive) or dependent target assumption (non-additive)

the value of the original network.[1] For example, if we adopt the network value as a function $f(\{n_i\}) = \sum_i n_i^2$, where n_i is number of nodes in the ith component, the value of the original network is $20^2 = 400$. After removing node 3, the network will be divided into two components: one 18-node network and one isolated node, the network value is reduced to $18^2 + 1^2 = 325$. Thus the benefit of node 3 is equal to the decrement $400 - 325 = 75$. Similarly, we can get the benefits of other nodes as illustrated in the bottom table of Fig. 2.2. In traditional security game models, they assume that the benefit of strategy {1, 2} and {3, 4} is equal to $39 + 39 = 78$ and $75 + 75 = 150$. The mixed strategy equilibrium[2] under this case is that defender choose nodes 1, 2 with probability 0.34 and nodes 3, 4 with probability 0.66. Instead, if we adopt the true value of nodes {1, 2} and {3, 4} (as illustrated in red of bottom table), the equilibria is that the defender chooses nodes 1, 2 with probability 0.63 and nodes 3, 4 with probability 0.37. From the point view of the network, the second one provides a more reliable strategy.

Based on the above example, we have the following observations: first, the traditional security game models do not consider dependency among the different targets; second, the attacker can attack at most one target. In particular, the payoff functions for both players are additive, i.e., the payoff of a group of targets is the sum of the payoffs of each target separately. This assumption means that the security agency measures the importance of several targets without considering the synergy among them. In practice, the attacker can simultaneously attack multiple targets and there exists some linkage structure among those targets such that attacking one target will influence the other targets. For example, an attacker attempts to destroy the

[1] Compared with traditional measures such as degree and betweenness centrality, the network value provides a more accurate description of the importance of different nodes.

[2] In this example, we adopt the zero-sum game model and assume the defender can protect the nodes with probability 1.

connectivity of a network and the defender aims to protect it. The strategy for each players is to choose the nodes of the network (to defend or to attack). If there are two nodes (node 1 and 2 in previous example) that constitute a bridge of this network, successfully attacking both of them will split the network into two parts and incur a huge damage, while attacking any one of them will have no significant impact. These observations show that proactive network defense introduces new challenges in computational game theory, and calls for the new theoretical development. The rest of this chapter mainly focus on how to develop a general game-theoretical path and algorithmic framework in proactive network defense.

2.2 Non-additive Security Game: A General Formulation of Network Security Game

Motivated by the previous example, we are now ready to define the non-additive security game (NASG) [15, 16].

Players and Targets The NASG contains two players (a *defender* and an *attacker*), and n targets. We use $[n] \triangleq \{1, 2, \ldots, n\}$ to denote the set of these targets. The attacker and defender need not be individuals, but could also be the organizations and groups who adopt a joint strategy. The target can be quite general and dependent on the application in mind. For example, they could represent links in the communication networks, roads in the urban networks or cities in the whole country.

Strategies and Index Function The *pure* strategy for each player is the subset of targets and all the pure strategies for each player constitute a collection of subsets of $[n]$. We assume that the attacker can attack at most c targets, where $c > 1$ is a constant. The attacker's pure strategy space is a uniform matroid $\mathscr{A} = \{A \subseteq [n] | |A| \leq c\}$ and the number of attacker's pure strategies is $N_a \triangleq |\mathscr{A}|$. Similarly, we use $\mathscr{D} \in 2^{[n]}$ to denote the defender's pure strategy space and $N_d \triangleq |\mathscr{D}|$. Note that there exists some resource allocation constraints in practice and such that $\mathscr{D}$ is not always a uniform matroid. For example, if the defender has a budget and its resource are obtained at some costs, in which the costs are heterogeneous. In this case, the defender's feasible pure strategy corresponds to all the possible combinations of the targets with total cost less than the budget.

Suppose that the order of the pure strategy of the attacker is given by index function $\sigma(\cdot)$, which is a one-one mapping: $2^{[n]} \rightarrow \{1, 2, \cdots, 2^n\}$. Then, we define the following index function $\mu(\cdot)$ for the pure strategy of the defender as: $\mu(U) = \sigma(U^c)$ for any $U \in 2^{[n]}$. For simplicity, the index function $\sigma(\cdot)$ and $\mu(\cdot)$ are defined over all subsets of $[n]$. The reason behind this definition of the index function is to simplify the representation of most of the theoretical results. For example, if $n = 2$, $\mathscr{A} = \mathscr{D} = 2^{\{1,2\}}$, and the order of the attacker's pure strategy is $\sigma(\{1, 2\}) = 1$, $\sigma(\{2\}) = 2$, $\sigma(\{1\}) = 3$ and $\sigma(\emptyset) = 4$, then the order for defender's pure strategy is $\mu(\emptyset) = 1$, $\mu(\{1\}) = 2$, $\mu(\{2\}) = 3$ and $\mu(\{1, 2\}) = 4$.

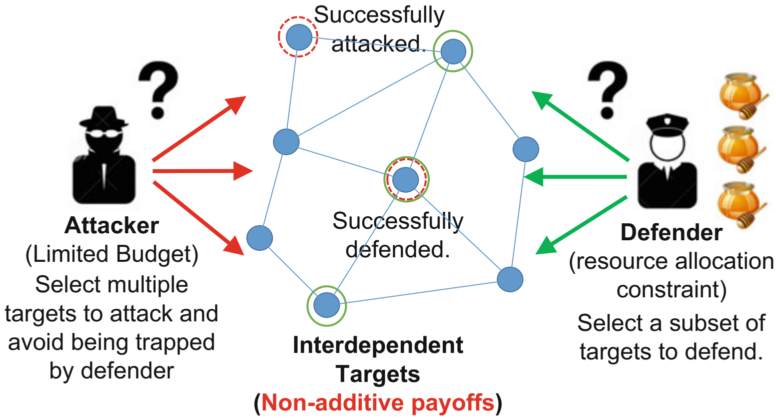

Successfully attacked.
?
Attacker
(Limited Budget)
Select multiple targets to attack and avoid being trapped by defender
Successfully defended.
Interdependent Targets
(Non-additive payoffs)
?
Defender
(resource allocation constraint)
Select a subset of targets to defend.

Fig. 2.3 Network security game with non-additive utility functions and multiple attacker resources

The *mixed* strategy is the probability distribution over the pure strategy space, which is employed when the player determines its strategy based on some random experiment. For example, if the attacker chooses $\mathbf{p}$ as its mixed strategy, the probability that strategy A is chosen is $\mathbf{p}_{\sigma(A)}$. The set of all the mixed strategies of the attacker and defender can be represented as the simplex Δ_{N_a} and Δ_{N_d}, where

$$\Delta_{N_a} = \{\mathbf{p} \in \mathbb{R}_+^{N_a} | \sum_{A \in \mathscr{A}} \mathbf{p}_{\sigma(A)} = 1\}. \tag{2.1}$$

A similar definition holds for Δ_{N_d}.

Payoff Structure The benefits and losses are represented by utility functions as follows. Let set function $B(\cdot) : \mathscr{A} \rightarrow \mathbb{R}$ and $L(\cdot) : \mathscr{A} \rightarrow \mathbb{R}$ be the attacker's benefit and loss functions, respectively. The standard assumption is that the benefit is always larger than the loss: $B(A) > L(A)$ for all $A \in \mathscr{A}$. If the attacker and defender choose strategy $A \in \mathscr{A}$ and $D \in \mathscr{D}$, the attacker's and defender's payoff is given by $B(A \backslash D) + L(A \cap D)$ and $-L(A \cap D) - B(A \backslash D)$, respectively.[3] In this payoff structure, one can see that the game is zero-sum such that one player's benefit is indeed the loss of the other players. For more complex non-zero sum games, please refer to [16].

Bilinear-Form Based on the above payoff structure, we can define the benefit matrices of attacker $\mathbf{B} : \forall A \in \mathscr{A}, D \in \mathscr{D}$,

$$\mathbf{B}_{\sigma(A),\mu(D)} = B_a(A \backslash D), \tag{2.2}$$

[3] $A \backslash D$ is the standard set difference, defined by $A \backslash D = \{x | x \in A, x \notin D\}$ and is equal to $A \cap D^c$, where D^c is the complementary set of subset D. An example of NASG is illustrated in (Fig. 2.3).

and the loss matrices: $\mathbf{L}$: $\forall A \in \mathscr{A}$, $D \in \mathscr{D}$,

$$\mathbf{L}_{\sigma(A),\mu(D)} = L_a(A \cap D). \tag{2.3}$$

Let $\mathbf{M}^a$ and $\mathbf{M}^d$ be the attacker's and defender's payoff matrices. It is clear that $\mathbf{M}^a = \mathbf{B} + \mathbf{L}$ and $\mathbf{M}^d = -\mathbf{B} - \mathbf{L}$. Then the expected payoffs for the attacker and defender are given by following bilinear form, when they play the mixed strategy $\mathbf{p} \in \Delta_{N_a}$ and $\mathbf{q} \in \Delta_{N_d}$, by

$$U_a(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T \mathbf{M}^a \mathbf{q} \quad \text{and} \quad U_d(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T \mathbf{M}^d \mathbf{q}. \tag{2.4}$$

Solution Concepts If both players move simultaneously, the standard solution concept is the *Nash equilibrium (NE)*, in which no single player can obtain a higher payoff by deviating unilaterally from this strategy. A pair of mixed strategies ($\mathbf{p}^*$, $\mathbf{q}^*$) forms a *NE* if and only if they satisfy the following: $\forall \mathbf{p} \in \Delta_{N_a}$, $\mathbf{q} \in \Delta_{N_d}$,

$$U_d(\mathbf{p}^*, \mathbf{q}^*) \geq U_d(\mathbf{p}^*, \mathbf{q}) \text{ and } U_a(\mathbf{p}^*, \mathbf{q}^*) \geq U_a(\mathbf{p}, \mathbf{q}^*). \tag{2.5}$$

In some application domain, the defender can build fortifications before the attack and is thus in the leader's position from the point view of the game, and able to move first. In this case, the *strong Stackelberg equilibrium (SSE)* serves as a more appropriate solution concept [17, 18], where the defender commits to a mixed strategy; the attacker observes this strategy and comes up with its best response(s). Formally, let $C(\mathbf{q}) = \arg\max_{\mathbf{p} \in \Delta_{N_a}} U^a(\mathbf{p}, \mathbf{q})$ denote the attacker's best response to defender's mixed strategy $\mathbf{q}$. A pair of mixed strategies ($\mathbf{p}^*$, $\mathbf{q}^*$) is a SSE, if and only if,

$$\mathbf{q}^* = \arg \max_{\mathbf{q} \in \Delta_{N_d}} U_d(C(\mathbf{q}), \mathbf{q}) \text{ and } \mathbf{p}^* = C(\mathbf{q}^*). \tag{2.6}$$

Our goal is to compute the defender's Nash equilibrium strategies and strong Stackelberg equilibrium strategies, and we call it the **equilibrium computation problem**.

2.3 Curse of Dimensionality and Compact Representation Technique

The Nash equilibrium is equivalent to the strong Stackelberg equilibrium in the zero-sum game. Therefore, we only need to focus on the computation of Nash equilibrium. Invoking the result in the von Neumann's minimax theorem, computing the NE of zero-sum game can be formulated as the following minimax problem,

$$\min_{\mathbf{q} \in \Delta_{N_d}} \max_{\mathbf{p} \in \Delta_{N_a}} U_a(\mathbf{p}, \mathbf{q}) = \mathbf{p}^T \left(\mathbf{B}^a + \mathbf{L}^a\right) \mathbf{q}. \tag{2.7}$$

One standard solution path is transforming the above problem into the following linear programming problem.

$$\begin{aligned} \min_{\mathbf{q},u} \quad & u \\ s.t. \quad & \mathbf{v}^T\left(\mathbf{B}^a+\mathbf{L}^a\right)\mathbf{q} \le u, \forall \mathbf{v} \in \Delta_{N_a}, \\ & \mathbf{q} \in \Delta_{N_d}. \end{aligned} \tag{2.8}$$

Curse of Dimensionality It is well known that the linear programming problem can be solved in polynomial time of number of variables and constraints by using the interior point method. However, the above linear programming problem contains N_d+1 number of variables and N_a+N_d constraints, which is at least the size of defender's pure strategy space. In the worst case, i.e., the defender can protect any subsets of targets and $N_a=\Theta(2^n)$. Moreover, unlike the traditional security game [10] that assumes that attacker only attack one target, there exists poly(n) number of variables and exponential number of constraints. One can use the cutting plane (ellipsoid method) to get a polynomial time reduction. However, in this problem, due to multiple attacker resources, it becomes a much more complicated issue, and calls for the development of a new theoretical path.

The goal of the rest of this subsection is to develop a technique to compactly and equivalently represent the zero-sum and non-additive security game with only poly(n) variables. To convey our idea more easily, we begin with an example.

We first use gauss elimination on matrices $\mathbf{B}^a$ and $\mathbf{L}^a$ to transform them into row canonical form, which is to left and right multiply such matrices by elementary matrices $\mathbf{E}_1, \mathbf{E}_2 \in \mathbb{R}^{N_a\times N_a}$ and $\mathbf{F}_1, \mathbf{F}_2 \in \mathbf{R}^{N_d\times N_d}$.

$$\begin{aligned} \min_{\mathbf{q}\in\Delta_{N_d}} \max_{\mathbf{p}\in\Delta_{N_a}} \mathbf{p}^T\left(\mathbf{B}^a+\mathbf{L}^a\right)\mathbf{q} &= \min_{\mathbf{q}\in\Delta_{N_d}} \max_{\mathbf{p}\in\Delta_{N_a}} \mathbf{p}^T\mathbf{E_1}\mathbf{E}_1^{-1}\mathbf{B}^a\mathbf{F}_1^{-1}\mathbf{F_1q} \\ &\quad + \mathbf{p}^T\mathbf{E}_2\mathbf{E}_2^{-1}\mathbf{L}^a\mathbf{F}_2^{-1}\mathbf{F}_2\mathbf{q} \\ &= \min_{\mathbf{q}\in\Delta_{N_d}} \max_{\mathbf{p}\in\Delta_{N_a}} \mathbf{p}^T\mathbf{E}_1\begin{bmatrix}\mathbf{B}_r^a & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{F}_1\mathbf{q} \\ &\quad + \mathbf{p}^T\mathbf{E}_2\begin{bmatrix}\mathbf{L}_s^a & \mathbf{0}\\ \mathbf{0} & \mathbf{0}\end{bmatrix}\mathbf{F}_2\mathbf{q}. \end{aligned}$$

where r and s are the rank of matrices $\mathbf{B}^a$, $\mathbf{L}^a$, and $\mathbf{B}_r^a$, $\mathbf{L}_s^a$ are the corresponding non-zero blocks of their row canonical form. If we define the affine transformation: $f_1(\mathbf{p})=\left(\mathbf{p}^T\mathbf{E}_1\right)^T$, $f_2(\mathbf{p})=\left(\mathbf{p}^T\mathbf{E}_2\right)^T$, $g_1(\mathbf{q})=\mathbf{F}_1\mathbf{q}$ and $g_2(\mathbf{q})=\mathbf{F}_2\mathbf{q}$. Let[4]

$$\begin{aligned} \Delta_{N_a}^a &= \{(f_1(\mathbf{p}), f_2(\mathbf{p}))|\mathbf{p}\in\Delta_{N_a}\}, \\ \Delta_{N_d}^d &= \{(g_1(\mathbf{q}), g_2(\mathbf{q}))|\mathbf{q}\in\Delta_{N_d}\}. \end{aligned}$$

we can obtain the following equivalent optimization problem,

$$\min_{(\bar{\mathbf{q}}_1,\bar{\mathbf{q}}_2)\in\Delta^d_{N_d}} \max_{(\bar{\mathbf{p}}_1,\bar{\mathbf{p}}_2)\in\Delta^a_{N_a}} \bar{\mathbf{p}}_1^T \begin{bmatrix} \mathbf{B}^a_r & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{q}}_1 + \bar{\mathbf{p}}_2^T \begin{bmatrix} \mathbf{L}^a_s & \mathbf{0} \\ \mathbf{0} & \mathbf{0} \end{bmatrix} \bar{\mathbf{q}}_2.$$

Moreover, considering the fact that only the first r elements in vector $\bar{\mathbf{p}}_1$ and $\bar{\mathbf{q}}_1$, and the first s elements in $\bar{\mathbf{p}}_2$ and $\bar{\mathbf{q}}_2$ have non-zero coefficients in the above optimization model, we can further simplify the above optimization problem as

$$\min_{(\bar{\mathbf{q}}_1,\bar{\mathbf{q}}_2)\in H_d} \max_{(\bar{\mathbf{p}}_1,\bar{\mathbf{p}}_2)\in H_a} \bar{\mathbf{p}}_1^T \mathbf{B}^a_r \bar{\mathbf{q}}_1 + \bar{\mathbf{p}}_2^T \mathbf{L}^a_s \bar{\mathbf{q}}_2, \tag{2.9}$$

where the H_a and H_d is obtained by projecting the polytope $\Delta^a_{N_a}$ and $\Delta^d_{N_d}$ to those coordinates belonging to the non-zero blocks.

The basic observation in the above example is that the number of variables in the optimization model (2.9) is equal to the sum of rank $r+s$ of payoff matrices. Based on the rank inequality that the rank of a matrix is less than its dimension, we have that $r, s \leq \min\{N_a, N_d\}$. Since the number of attacker's pure strategies is $N_a = O(n^c) = \text{poly}(n)$. Therefore, there exists at most poly(n) variables in the optimization model (2.9).

The above illustrative derivation provides a possible path to compactly represent the game. However, there exists a significant technical challenge: the elementary matrices $\mathbf{F}_1$, $\mathbf{F}_2$ and their inverse matrices may have an exponential size due to the exponentially large defender's pure strategy space. Hence, the key question is whether we can **find both these elementary matrices** efficiently? To tackle this problem, we first show that payoff matrices $\mathbf{B}^a$ and $\mathbf{L}^a$ can be decomposed as the product of the several simple matrices.

Theorem 2.1 (Decomposition of the Payoff Matrix)
The payoff matrix $\mathbf{M}^a = \mathbf{B}^a + \mathbf{L}^a$ *can be decomposed as*

$$\mathbf{M}^a = \mathbf{E}(\mathbf{D}^b\mathbf{J} + \mathbf{D}^l\mathbf{K}), \tag{2.10}$$

where $\mathbf{D}^b, \mathbf{D}^l \in \mathbb{R}^{N_a\times N_a}$ *are the diagonal matrices with*

$$\mathbf{D}^b_{\sigma(A),\sigma(A)} = B^c(A), \mathbf{D}^l_{\sigma(A),\sigma(A)} = L^c(A), \forall A \in \mathscr{A}.$$

The $\mathbf{E} \in \mathbb{R}^{N_a\times N_a}$ *and* $\mathbf{J}, \mathbf{K} \in \mathbb{R}^{N_a\times N_d}$ *are binary matrices:*

(continued)

[4] The notation $(\cdot, \cdot)$ denotes the concatenation operator of vector.

$$\mathbf{E}_{\sigma(A),\sigma(U)} = \mathbf{1}\{U \subseteq A\}, \forall A, U \in \mathscr{A}$$
$$\mathbf{J}_{\sigma(A),\mu(D)} = \mathbf{1}\{A \subseteq D^c\},$$
$$\mathbf{K}_{\sigma(A),\mu(D)} = \mathbf{1}\{A \subseteq D\}, \forall A \in \mathscr{A}, D \in \mathscr{D}.$$

The common utility is defined as the Möbius transformation [19, 20] of the benefit and loss function $B(U)$ and $L(U)$ for all $U \in 2^{[n]}$,

$$\begin{aligned} B^c(U) &= \sum_{V \subseteq U} (-1)^{|U \setminus V|} B_a(V) \\ L^c(U) &= \sum_{V \subseteq U} (-1)^{|U \setminus V|} L_a(V). \end{aligned} \tag{2.11}$$

As can be seen in Theorem 2.1, we decompose the original exponentially large payoff matrix $\mathbf{M}^a$ into the summation and the product of several simple matrices including binary matrices $\mathbf{E}$, $\mathbf{J}$, $\mathbf{K}$ and two polynomial-sized diagonal matrices $\mathbf{D}^b$ and $\mathbf{D}^l$. Moreover, such a decomposition has a closed-form expression and the elements in those simple matrices can be implicitly represented.

Based on the above decomposition results, we can let the elementary matrices $\mathbf{E}_1 = \mathbf{E}_2 = \mathbf{E}$, $\mathbf{F}_1 = \mathbf{J}$ and $\mathbf{F}_2 = \mathbf{K}$, and the corresponding affine transformation $f(\mathbf{p}) = \mathbf{E}^T\mathbf{p}$ and $g_1(\mathbf{q}) = \mathbf{J}\mathbf{q}$, $g_2(\mathbf{q}) = \mathbf{K}\mathbf{q}$ to yield two polytopes: $\Delta^a_{N_a} = \{f(\mathbf{p})|\mathbf{p} \in \Delta_{N_a}\}$ and $\Delta^d_{N_d} = \{(g_1(\mathbf{q}), g_2(\mathbf{q}))|\mathbf{q} \in \Delta_{N_d}\}$. Then we can represent the minimax problem (2.7) as

$$\min_{(\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2) \in \Delta^d_{N_d}} \max_{\bar{\mathbf{p}} \in \Delta^a_{N_a}} \bar{\mathbf{p}}^T(\mathbf{D}^b\bar{\mathbf{q}}_1 + \mathbf{D}^l\bar{\mathbf{q}}_2), \tag{2.12}$$

The following definitions are often used in our next step theoretical development.

Definition 2.1 (Support Set) The support set of the non-additive security game is defined as

$$S = \{A \in \mathscr{A} | B^c(A) \neq 0 \text{ or } L^c(A) \neq 0\}. \tag{2.13}$$

and the support index set $\sigma(S) = \{\sigma(A)|A \in S\}$.

Definition 2.2 (Projection Operator) The projection operator $\pi_S : \mathbb{R}^N \to \mathbb{R}^{|S|}$ is

$$\pi_S((\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N)) = (\ldots, \mathbf{x}_i, \ldots)_{i \in \sigma(S)}, \tag{2.14}$$

and projection of polytope: $\Pi_S(\Delta_N) \triangleq \{\pi_S(\mathbf{x})|\mathbf{x} \in \Delta_N\}$.

Based on the definition of our support set S and matrices $\mathbf{D}^b$, $\mathbf{D}^l$, only the variables with indices belonging to $\sigma(S)$ have non-zero coefficients. Therefore, we can eliminate those variables with zero coefficients in (2.12) and project the polytopes $\Delta^a_{N_a}$ and $\Delta^d_{N_d}$ into the coordinates with indices belonging to $\sigma(S)$. The further simplified model can be expressed as

Compact Minimax Problem

$$\min_{(\bar{\mathbf{q}}_1,\bar{\mathbf{q}}_2)\in H_d}\ \max_{\bar{\mathbf{p}}\in H_a} \bar{\mathbf{p}}^T(\widetilde{\mathbf{D}}^b\bar{\mathbf{q}}_1+\widetilde{\mathbf{D}}^l\bar{\mathbf{q}}_2), \tag{2.15}$$

where[5] $H_a = \Pi_S(\Delta^a_{N_a})$, $H_d = \Pi_S(\Delta^d_{N_d})$, matrix $\widetilde{\mathbf{D}}^b$ and $\widetilde{\mathbf{D}}^l$ is obtained by extracting the non-zero columns and rows of matrix $\mathbf{D}^b$ and $\mathbf{D}^l$.

Since the size of the support set $|S| \leq N_a$, and $N_a = \text{poly}(n)$, we arrive at a compact representation of the non-additive security game with only poly(n) variables. Note that in the above compact representation framework, the affine transformation f_1 and f_2 are the same as in our compact representation. The following theorem guarantees the correctness of our compact representation.

Theorem 2.2 (Compact Representation) ($\mathbf{p}^*$, $\mathbf{q}^*$) *is a Nash equilibrium of zero-sum non-additive security game if and only if* $(\pi_S(f(\mathbf{p}^*)), (\pi_S(g_1(\mathbf{q}^*)), \pi_S(g_2(\mathbf{q}^*)))$ *is the optimal solution of compact minimax problem* (2.15).

2.4 Oracle-Based Algorithmic Framework

In the previous section, we develop a compact representation technique such that one can equivalently represent the original NASG by a minimax problem with a polynomial number of variables, which can be further solved by the following linear programming model,

Compact Linear Programming

$$\begin{aligned} &\min \quad u \\ &s.t. \quad \mathbf{v}^T(\widetilde{\mathbf{D}}^b\bar{\mathbf{q}}_1+\widetilde{\mathbf{D}}^l\bar{\mathbf{q}}_2) \leq u, \forall \mathbf{v} \in I_a, \\ &\qquad\ (\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2) \in H_d, \end{aligned} \tag{2.16}$$

where I_a denotes the set of vertices of the convex polytope H_a. The above linear programming problem has poly(n) number of variables and potentially exponential number of constraints (due to the membership constraint $(\bar{\mathbf{q}}_1, \bar{\mathbf{q}}_2) \in H_d$). This motivates us to utilize the ellipsoid method to solve the problem.

[5] Note that each vector in $\Delta^d_{N_d}$ is consists of two parts $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$. Here the corresponding low-dimensional point is $(\pi_S(g_1(\mathbf{q}), \pi_S(g_2(\mathbf{q}))$.

2.4.1 Preliminaries

Let H be a non-empty convex polytope in $\mathbb{R}^n$. Given a vector $\mathbf{w} \in \mathbb{R}^n$, one wants to find a solution to $\max_{\mathbf{x} \in H} \mathbf{w}^T\mathbf{x}$. By “linear optimization over H”, we mean solving the problem $\max_{\mathbf{x} \in H} \mathbf{w}^T\mathbf{x}$ for any $\mathbf{w} \in \mathbb{R}^n$. A separation problem for H is that, given a vector $\mathbf{x} \in \mathbb{R}^n$, decide if $\mathbf{x} \in H$, and if not, find a hyperplane which separates $\mathbf{x}$ from H. The following results are due to Grötschel et al. [21].

Theorem 2.3 (Separation and Optimization) *Let $H \in \mathbb{R}^n$ be a convex polytope. There is a* poly *(n) time algorithm to solve the linear optimization problem over H if and only of there is a* poly *(n) time algorithm to solve the separation problem for H.*

Theorem 2.4 (Separation and Convex Decomposition) *Let $H \in \mathbb{R}^n$ be a convex polytope. If there is a* poly *(n) time algorithm to solve the separation problem for H, then there is a* poly *(n) time algorithm that, given any $\mathbf{x} \in H$, yields $(n+1)$ vertices $\mathbf{v}^1, \ldots, \mathbf{v}^{n+1} \in H$ and convex coefficients $\lambda_1, \ldots, \lambda_{n+1}$ such that $\mathbf{x} = \sum_{i=1}^{n+1} \lambda_i \mathbf{v}^i$.*

2.4.2 Reduction Between NASG and Combinatorial Optimization

The main result in this subsection is captured in the following theorem.

Theorem 2.5 (NE Computation and Defender Oracle Problem)
There is a poly (n) *time algorithm to compute the defender's Nash equilibrium (strong Stackelberg equilibrium),* ***if and only if*** *there is a* poly (n) *time algorithm to compute the defender oracle problem: for any given vector $\mathbf{w} \in \mathbb{R}^{2|S|}$, determine,*

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \in I_d} \mathbf{w}^T \mathbf{x}. \tag{2.17}$$

To obtain above reduction, we adopt the following path: we first show how the compact problem and the defender oracle problem can be reduced to each other in poly(n) time; then we exploit the geometric structure of polytope H_a and H_d to construct two poly(n) time vertex mapping algorithms to obtain the reduction between the equilibrium computation and the compact problem. This whole procedure also produces an algorithmic framework to the solve the NASG.

The polynomial time reduction between the defender oracle problem and the compact linear programming problem can be easily obtained by the ellipsoid method. The key lies in how to obtain the reduction between the equilibrium computation (2.7) and the compact linear programming problem. Actually, there exist two issues: first, how to transform the input instance of each problem to the other one in poly(n) time; second, how to map the optimal solution of each problem to the other in poly(n) time. Since the input of the equilibrium computation problem are the utility functions $\{B(U)\}$ and $\{L(U)\}$ and the input of compact problem are the common utilities $\{B^c(U)\}$ and $\{L^c(U)\}$ (all the elements of matrices $\mathbf{D}^b$ and $\mathbf{D}^l$ are the common utilities), such transformation can be completed in $O(2^c n^c) =$ poly(n) time based on the definition of common utilities.

To resolve the second issue, we first consider how to map the optimal solution of compact problem to the defender's optimal mixed strategies. Based on Theorem 2.4, we obtain that if the separation problem of LP (2.16) can be solved in poly(n) time, we can decompose any feasible point $\mathbf{x}$ into a convex combination of at most $(2|S|+1)$ vertices of the polytope defined by those constraints. Note that this is precisely the DOP required for above reduction. Applying this result to the optimal solution $(\mathbf{q}_1^*, \mathbf{q}_2^*)$ of the LP (2.16), we can get a convex decomposition that

$$(\mathbf{q}_1^*, \mathbf{q}_2^*) = \sum_{i=1}^{2|S|+1} \lambda_i (\mathbf{v}_1^i, \mathbf{v}_2^i), \tag{2.18}$$

where $(\mathbf{v}_1^i, \mathbf{v}_2^i) \in I_d$. The basic fact is that the defender's mixed strategy can be regarded as a convex combination of its pure strategies, each of which corresponds to a vertex of simplex Δ_{N_d}. If we can map the vertices $(\mathbf{v}_1^i, \mathbf{v}_2^i)$ back to the vertices (pure strategy) of the original game, denoted by $h((\mathbf{v}_1^i, \mathbf{v}_2^i))$, the mixed strategies of the defender can be expressed as

$$\mathbf{q}^* = \sum_{i=1}^{2|S|+1} \lambda_i h((\mathbf{v}_1^i, \mathbf{v}_2^i)). \tag{2.19}$$

Thus, the key lies in how to compute $h((\mathbf{v}_1^i, \mathbf{v}_2^i))$ in poly(n) time.

To tackle this problem, we need to investigate the geometric structure of polytope H_d. First, considering an arbitrary defender's pure strategy $D \in \mathscr{D}$, the corresponding vertex in Δ_{N_d} is a unit vector $\mathbf{e}^D \in \mathbb{R}^{N_d}$ with only one non-zero element $\mathbf{e}^D_{\mu(D)} = 1$. Based on the definition of the transformation $g_1(\mathbf{q})$ and $g_2(\mathbf{q})$, the corresponding point of polytope H_d is

$$(g_1(\mathbf{e}^D), g_2(\mathbf{e}^D)) = (\mathbf{J}\mathbf{e}^D, \mathbf{K}\mathbf{e}^D) = (\mathbf{J}_{\mu(D)}, \mathbf{K}_{\mu(D)}), \tag{2.20}$$

where $\mathbf{J}_{\mu(D)}$ and $\mathbf{K}_{\mu(D)}$is the $\mu(D)$th column of matrix $\mathbf{J}$ and $\mathbf{K}$. Then the corresponding point $\mathbf{v}^D$ of the projected polytope H^d is

$$\mathbf{v}^D = \left(\pi_S(\mathbf{J}_{\mu(D)}), \pi_S(\mathbf{K}_{\mu(D)})\right), \tag{2.21}$$

which is the sub-vector of $\mathbf{J}_{\mu(D)}$ and $\mathbf{K}_{\mu(D)}$. The problem is that the vertex in the high-dimensional polytope may not project to a vertex of its low-dimensional image. However, the following lemma will provide a positive result.

Lemma 2.1 (Geometric Structure of H_d)
For any support set $[n] \in S \in \mathscr{A}$, the vertices of the polytope H_d are the columns of the sub-matrix of $\begin{bmatrix} \mathbf{J} \\ \mathbf{K} \end{bmatrix}$, which is formed by extracting the row whose index belongs to $\sigma(S)$.

Since we have a closed-form expression of the matrix **J** and **K**, we can construct a vertex mapping algorithm from low-dimensional vertex to the defender's pure strategy. The efficiency and the correctness of Algorithm 1 is justified by following lemma.

Algorithm 1: Vertex mapping from vertex to pure strategy

```
input  : Vertex $(\mathbf{v}_1, \mathbf{v}_2) \in I^d$.
output: Defender's pure strategy $D$.
$T = \emptyset$;
for each $i \in [n]$ do
    Examine each coordinate of vertex:
    if $\mathbf{v}_{1,\sigma(\{i\})} \neq 0$ then
        $T = T \cup \{i\}$;
    end
end
$D = T^c$;
```

Lemma 2.2 (Correctness of Vertex Mapping Algorithm) *The vertex mapping Algorithm* 1 *runs in $O(n)$ time and maps each vertex of H_d to a unique pure strategy.*

Note that our vertex mapping algorithm only examines n instead of all the coordinates of each vertex of H_d to recover a defender's pure strategy. The reason behind this result is that there exists a one-to-one correspondence between each pure strategy and those n coordinates of each vertex of polytope H_d. Intuitively, those n coordinates of each vertex of H_d is binary and therefore there exists possibly 2^n possibilities, each of which corresponds to a pure strategy.

The other direction follows from the following argument. Suppose that the problem of equilibrium computation is solved in poly (n) time and the optimal defender's mixed strategy is denoted by $\mathbf{q}^*$. Invoking a known result in game theory (Theorem 4 in [22]), the support size, i.e., number of strategies with nonzero probability, of the Nash equilibrium is less than the rank of the payoff matrix. Since the rank of payoff matrix $\mathbf{M}^a$ is $O(n^c)$, the number of non-zero coordinates in $\mathbf{q}^*$ is at most $O(n^c) =$ poly(n) and $\mathbf{q}^*$ can be expressed as

$$\mathbf{q}^* = \sum_{i=1}^{\text{poly}(n)} \lambda_i \mathbf{e}^i. \tag{2.22}$$

Therefore, we can determine the optimal solution of the compact problem in poly(n) time by constructing the following poly(n) time vertex mapping algorithm from a pure strategy $\mathbf{e}^i$ to a vertex of H_d.

Algorithm 2: Vertex mapping from pure strategy to vertex

```
input  : Defender's Pure Strategy $D$.
output: Vertex $\mathbf{v}^D \in I_d$
$T = \emptyset$;
for each $V \in \mathscr{A}$ do
    if $V \subseteq D^c$ then $\mathbf{v}^D_{1,\sigma(V)} = 1$;
    else $\mathbf{v}^D_{2,\sigma(V)} = 0$;
end
Output vertex $\mathbf{v}^D = (\mathbf{v}^D_1, \mathbf{v}^D_2)$;
```

The intuition behind this result is similar to the previous vertex mapping algorithm and the correctness of Algorithm 2 is guaranteed by the following lemma.

Lemma 2.3 (Correctness of Vertex Mapping Algorithm)
Vertex mapping Algorithm 2 runs in $O(n^c)$ time and maps each defender's pure strategy D to a unique vertex of H_d.

Combining all the above results together, we provide a general algorithmic framework shown next.

Algorithm 3: General algorithmic framework for non-additive security game

```
1. Utility transformations: Transform the original utility functions $\{B(U)\}$ and $\{L(U)\}$ to the
   corresponding common utilities $\{B^c(U)\}$ and $\{L^c(U)\}$ based on Möbius transformation;
2. Solve the compact problem: Solve the linear program (2.16) to obtain the optimal
   compact strategy $\mathbf{t}^*$ by ellipsoid method;
3. Convex decomposition: Decompose optimal compact strategy $\mathbf{t}^*$ into the convex
   combination: $\mathbf{t}^* = \sum_{i=1}^{n+1} \lambda_i \mathbf{v}^i$ by exactly solving the defender oracle problem;
4. Vertex mapping: Map each vertex $\mathbf{v}^i$ to a defender pure strategy $D_i$ by Algorithm 1,
   output the defender's NE strategy:
     play pure strategy $D_i$ with probability $\lambda_i$, $1 \le i \le n + 1$;
```

2.4.3 Applications

In this subsection, we will discuss the applications of our developed algorithmic framework to several security domain problems.

2.4.3.1 Network Security Game

The network security game [14, 23] is given by the following definitions.

Definition 2.3 A network security game is given by the tuple (G, T, $\mathbf{F_a}$, c), where $G = (V, E)$ with node set V, edge set E, T is the network value function, $\mathbf{F_a}$ is the failure operator, c is the maximum number of nodes the attacker can choose, while the defender can protect any target.

The network value function $T : G \to \mathbb{R}$ is a security measure assessing the utility of a network, and failure operator $\mathbf{F_a} : 2^G \to 2^G$ is to generate a new network via a specific failure mode after removing some nodes. For example, Shakarian et al. [23] adopt the number of connected load nodes as T, and edge cascading failure model as $\mathbf{F_a}$. We next discuss several classical network security games *that can be solved in polynomial time*.

Example 2.3 (Security Game in a Tree Network) In cybersecurity, the sensor network often exhibits a tree topology. The game is such that the attacker attempts to invade some nodes to destroy the connectedness of the network and the IT manager is required to deploy anti-virus software in some nodes. Suppose that the network G consists of m connected components: $V_1, V_2, \ldots, V_m$ and both players adopt the following network value functions

$$T(G) = \max_{1 \le i \le m} |V_i|. \tag{2.23}$$

In practice, we assume that the attacker can simultaneously invade at most two nodes, i.e., $c = 2$. Then, if node i is attacked, the tree G is divided into 2 sub-trees: G_{i1} and G_{i2}, and the benefit is given by

$$B(\{i\}) = n - \max\{|G_{i1}|, |G_{i2}|\} = \min\{n - |G_{i1}|, n - |G_{i2}|\}.$$

Similarly, if node j is attacked, the tree G is divided into 2 sub-trees: G_{j1} and G_{j2}, and the benefit is given by

$$B(\{j\}) = n - \max\{|G_{j1}|, |G_{j2}|\} = \min\{n - |G_{j1}|, n - |G_{j2}|\}.$$

Without loss of generality, suppose $j \in G_{i2}$, then if nodes i, j are simultaneously attacked, the tree G is divided into 3 subtrees: G_{i1}, G_{i21} and G_{i22}, where the latter two are obtained by dividing G_{i2}. The corresponding benefit is given by

$$B(\{i,j\}) = n - \max\{|G_{i1}|, |G_{i21}|, |G_{i22}|\} = \min\{n - |G_{i1}|, n - |G_{i21}|, n - |G_{i22}|\}.$$

Then one can easily show that the following holds true for any $i, j \in [n]$,

$$B(\{i, j\}) \leq B(\{i\}) + B(\{j\})$$

and $B^c(\{i, j\}) \leq 0$. Combining this result with Theorem 2.5, one can easily show that the defender oracle problem is a submodular minimization problem, which can be solved in polynomial time. Further, we can use Algorithm 1 to determine an equilibrium strategy in polynomial time.

Example 2.4 (Security Game in a Sparse Network) As can be seen in (Fig. 2.4), the real world network is extremely sparse and the largest connect component is always small compared to the network scale, i.e, $O(\log(n))$. In this case, we have the following result.

Lemma 2.4 *A network security game* $(G, T, \mathbf{F_a}, c)$ *can be solved in* poly (n) *time if the largest connected component of* G *is* $\Theta(\log(n))$.

The basic intuition is that, when the network is extremely sparse such that the largest connected component of G is $\Theta(\log(n))$, the common utility functions defined in (2.11) will satisfy a separable condition

$$U = \bigcup_{i=1}^{m} U_i, \forall U_i \subset V_i, U_i \neq \emptyset$$
$$\Longrightarrow B^c(U) = C_a^c(U) = C_d^c(U^c) = 0.$$

Then, one can easily show that the defender oracle problem can be separated into $O(n)$ subproblems, each of which can be solved in polynomial time. Combining this result with Theorem 2.5, we can solve this network security game in polynomial time.

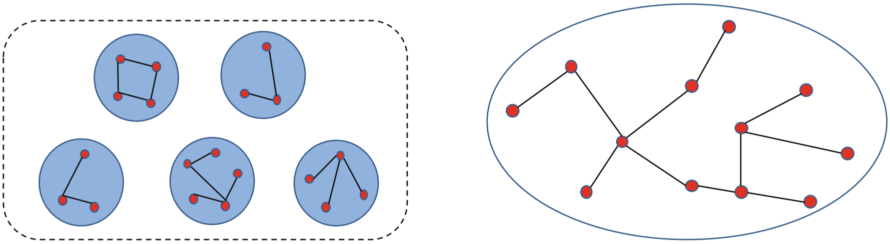


Fig. 2.4 Security game in a sparse network and tree network

2.4.3.2 Security Game with Multiple Attacker Resources

There exists several other important applications of our developed algorithmic framework.

LAX Airport Checkpoint Placement Problem [24] This problem is one of the earliest applications of security games. In this setting, the security force has k police officers that are to be deployed across n (where $k < n$) checkpoints. Each police officer can be deployed at any given check point. Therefore, any subset of $[n]$ of size at most k is a defender pure strategy. Korzhyk et al. [25] extends this game model into the multiple attacker resources and shows that this problem can still be solved in poly(n) time by a state transition algorithm [25]. In our framework, the DOP is the linear optimization over a uniform matroid.

$$\begin{aligned} \max \quad & \mathbf{w}^T \mathbf{x} \\ s.t. \quad & \textstyle\sum_{i=1}^{n} \mathbf{x}_i \le k, \mathbf{x} \in \{0, 1\}^n. \end{aligned} \tag{2.24}$$

The above problem can be solved in polynomial time by summing the k largest elements of vector $\mathbf{w}$. Thus, it verifies previous results.

In the following three cases, the defender's resources are heterogeneous such that there exists some practical constrains in the set system ε.

Geographic Constrained Patrolling Problem In the patrolling problem, due to geographic constraints, the police officer can only patrol the area around the station. In this case, the resources of different defenders (police) can defend different groups of targets. In our framework, we can construct a weighted bipartite graph as follows: (1) two disjoint sets U, V, where U represents all the nodes, and V represents all the resources; (2) there exists an edge between the node u in U and node v in V if the resource v can cover node u; (3) associate each edge (u,v) with a weight $\mathbf{w}_u$ ($\mathbf{w}$ is the vector in the DOP). Then the DOP is a weighted bipartite matching problem, which can be solved in polynomial time by Hungarian algorithm.

Federal Air Marshal Scheduling Problem [10] In such applications, one air marshal is assigned to protect several sequential flights with the constraint that any destination of the previous flight is the departure of the next flight. The objective is to cover all current flights. In [26], the authors investigate this problem under single attacker resources and shows the polynomial solvability in some cases and NP-hardness in other cases. However, attackers may initiate simultaneous attacks (e.g., the flights of 911) and there still does not exist any efficient algorithm. In our framework, we can construct the following weighted set cover problem: let the node set $[n]$ be the universe and all the air marshals constitute the collection S of subsets of $[n]$; then associate the weight $\mathbf{w}$ to each element of the universe. Then, the DOP is a weighted set cover problem and our results show that when the attacker has multiple resources, the problem is generally NP-hard but we can still solve this problem in some cases. For example, if each air marshal can protect at most two flights (a pair of round trip flights), the set system ε indeed encodes the weighted 2-cover, which can be solved in poly(n) time.

Spatio-Temporal Security Game [12,27] In many applications of security games, an important class is the spatio-temporal security game. This kind of game is used to model the games played in the spatio-temporal spaces such as planning patrol boats of the US Coast Guard [12], wildlife protection [27]. The current solution technique of this game is to discretize the space and time and build 2-D gird, in which the security force patrol the points. Combining the results in [28], we can show that spatio-temporal security game with multiple attacker resources are indeed a min-cost flow problem, which can be solved in poly(n) time.

There exist other applications that can be cast in our framework such as passenger screening for the Transportation Security Administration [11]. Indeed, based on our general framework in Algorithm 3, all the results under the single attacker resources can be directly extended to the scenario of multiple attacker resources.

2.5 Approximated Equilibrium Computation by Low Rank Decomposition

In the previous section, we have developed a compact representation technique and algorithmic framework such that one can reduce the problem of determining the equilibrium point of NASG to a combinatorial optimization problem. However, one pessimistic result is that the defender oracle problem in general is NP-hard, which is high-complexity to be solved in practice. A natural question is the following: in practical network security games, can we still efficiently solve an equilibrium point. Actually, one crucial observation is that *the common utility in realistic networks is concentrated around zero.*

In Fig. 2.5, we examine the distributions of the benefit function and its common utility function in the following two kinds of network: Erdös-Renyi network $G(n, p)$ and scale-free network $G(n, \alpha)$, where n is the number of nodes, p is the probability that any two nodes are connected, α is the parameter of degree distribution of the scale-free network. Suppose that the network G consists of m connected components: V_1, V_2, ..., V_m and we adopt the following two kinds of network value functions,

$$T_1(G) = \max_{1 \le i \le m} |V_i|, T_2(G) = \sum_{i=1}^{m} |V_i|^2.$$

The different form of network value functions have different assessment of the network. The detailed comparison can be found in [14]. As can be seen in Fig. 2.5, in both Erdös-Renyi and scale-free networks, although the distribution of the benefit function is random, the distribution of the common utility function is concentrated around zero and 90% of them are less than 0.05. In particular, when the number of

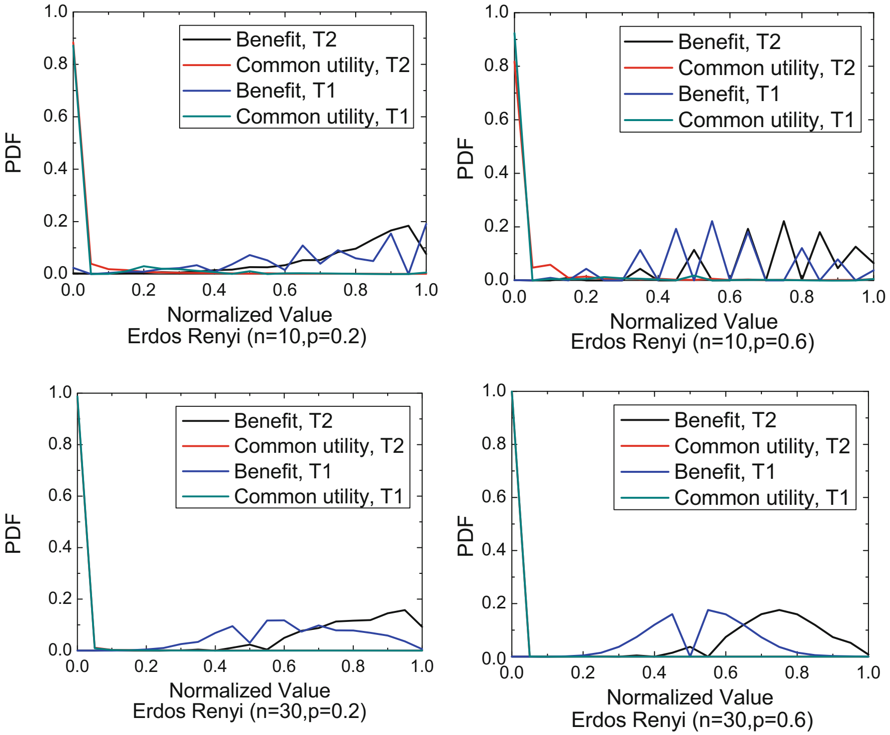


Fig. 2.5 The distributions of common utility function and benefit function. All their value are absolute value and normalized in [0, 1]

nodes increases, this phenomenon is amplified such that almost 99% of the common utility functions are less than 0.05.

Based on the above observation, we can let most of the common utility functions equal to 0 according to a given threshold ε_c. Formally, let $\tilde{B}^c(\cdot)$ denote the new common utility function generated by Algorithm 2, then the corresponding approximate benefit function satisfies

$$|\tilde{B}(U) - B(U)| = \left| \sum_{W \subseteq U} \tilde{B}^c(W) - \sum_{W \subseteq U} B^c(W) \right|$$
$$\leq \sum_{W \subseteq U} \left| \tilde{B}^c(W) - B^c(W) \right| \leq 2^{|U|} \epsilon_c.$$

Since $|U| \leq c$, the maximum error between the original benefit functions and new generated benefit functions is less than $2^c \varepsilon_c$. A classic result of game theory is that, if the maximum difference between the elements of two payoff matrices is bounded by ε, the difference of the optimal game values yielded by these two payoff matrices

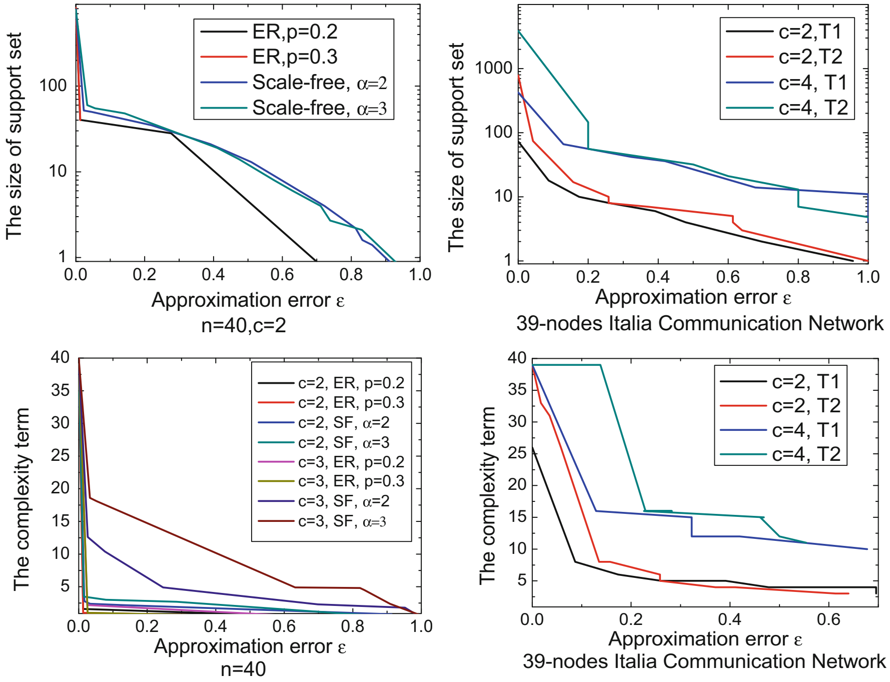


Fig. 2.6 Top: the size of support set $|S|$ versus approximation error ε; Bottom: the complexity term $\max_i |U_i|$ versus approximation error ε. Remark that the ε represents the approximation error of the game value. Note that SF denotes the scale-free network

are bounded by 2ε [22]. Therefore, the approximation error of our game value is bounded by $2^{c+1}\varepsilon_c$.

As shown at the top of Fig. 2.6, for the Erdös Renyi, scale-free and Italian communication network, the size of support set will be reduced 90% by an extremely small approximation error 0.05. Moreover, this process also leads to a separable structure of S, and the resulting complexity of solving the NASG is poly$(n)O(2^{\max_i |U_i|})$. For example, in the bottom of Fig. 2.6, the complexity term $\max_i |U_i|$ can be greatly reduced to the order of $\Theta(\log(n))$ with an approximation error of 1%, regardless of the size and density of the network, and how many targets the attacker can choose. More comprehensive numerical results can be found in [15]. In summary, our approximation framework can reduce the complexity term $\max_i |U_i|$ to order $\Theta(\log(n))$ by only 10% approximation error in most networks including Erdös-Renyi, scale-free network and a 39-nodes Italian communication network. Therefore, using our theoretical framework, we can **approximately and compactly represent a realistic network security game and solve it in poly(n) time with high accuracy**.

2.6 Future Research Directions

In this section, we outline several future research directions.

2.6.1 Learning-Based Proactive Network Defense

In our proposed NASG, we suppose that probability distribution of attacker type is known to the defender and regarded as a prior belief of defense group, and can be formed by the Bayesian rule. However, in practical settings, some of the information might be unknown to the defense group. This problem can be investigated by incorporating a learning framework into our Non-additive Security Game based on the following two scenarios: (1) full information setting: both the attacker type and action is known in each time slot t. We need to construct an online learning algorithm to form the belief of attacker type distribution; (2) partial information setting: only the attacker action is known in each time slot t. This kind of problem can be cast into a multi-arm bandits setting. The key challenge here will be in designing algorithms that provide a small (sublinear) regret.

2.6.2 Game-Theoretical Network Defense with Boundedly Rational Players

In our proposed NASG, we suppose that all the players are fully rational. However, in real situations, the players such as civilians will have bounded rationality. To model this behavior, the quantal response equilibrium is a more appropriate solution concept. The challenge is that, due to the introduction of the quantal response model, such an optimization problem has a non-convex fractional objective function, which is generally hard to solve. The goal lies in how to transform such an problem into a sequence of convex optimization problem and solve each sub-problem efficiently.

2.6.3 Multi-Scale Proactive Network Defense

In the previous sections, we have already discussed the general model and algorithmic framework of game-theoretical proactive network defense. However, in the future battlefields, there exists multiple factors that will greatly change the current game structure. For example, internet of things make the network structure highly dynamic. Other key factors includes:

Multi-Party Games In this game there exist multiple players in both the defense and adversarial groups. In addition, in practice, there may also exist “neutral players” that could potentially be influenced by the strategies of the attackers and defenders. These kinds of dependency are sometimes characterized by the underlying social networks formed among all the players. For example, in the battle with adversarial group Lashkar-e-Taiba, the players in the defense group would include the US and Indian Governments, as well as other peaceful nationals. They share the defense resources and cooperate with each other. In contrast, the players in the opposing groups include training camps, military bases, and get political supports from diaspora and foreign states. The players in the neutral group can be regarded as civilians or the weak peaceful groups in Pakistan.

Multi-Genre Networks In real scenarios, there exists some linkage structure among different infrastructures due to the effect of the underlying multi-genre networks. One well-known example is the interdependence network formed by power grid and communication systems [29]. Due to the dependencies among different targets, attacking one target will influence other targets. For instance, an attacker attempts to destroy the connectivity of a network and the defender aims to protect it. The strategy for both players is to choose the nodes of the network to either protect or attack. If there are two nodes that constitute a bridge in this network or inter-dependent network, successfully attacking both of them will split the network into two parts and incur a huge damage, while attacking any one of them may have limited impact.

Actually, as shown in Fig. 2.7, we can generalize NASG to Multi-Stage Multi-Party Bayesian Security Game with considering the interaction between multi-genre networks, multi-parties and uncertain attacker behavior. It contains a time horizon $T = \{1, 2, \ldots, t\}$ and runs Multi-Party Bayesian Security Games in each time slot t, which contains three kinds of players: defenders, attackers and neutrals. Social links could exist among some of the players in different groups such that the decision making of different players are dependent on each other. Each player i in the adversarial group is from a set of possible types θ_i (multiple adversary types trying to infiltrate security). The defender has a belief $p[t]$ of the attacker’s uncertainty, which is a probability distribution over all the adversarial players’ types. The belief $p[t]$ is a prior of defender before playing the game in time slot t, and can be formed by a Bayesian rule and learning the actions of all the agents in the previous time slots. The objective of the MMBSG is to calculate the mixed strategy (a probability distribution over each pure strategies) Nash Equilibrium (NE) in each time slot t, and the key lies in how the solve this game efficiently based on our previously developed technique.

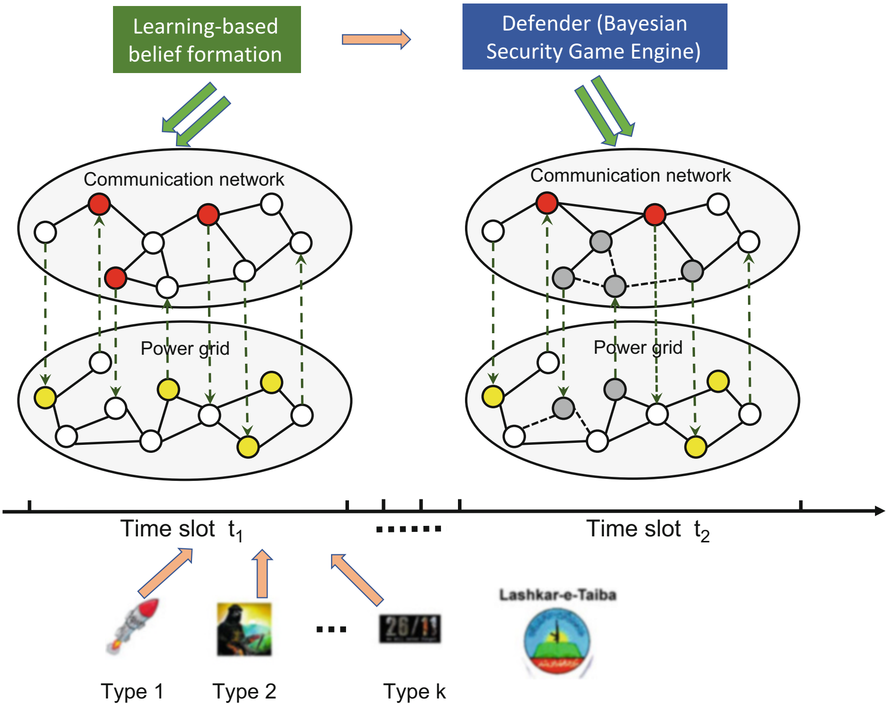


Fig. 2.7 Overview of our proposed compositional game theory framework consisting of an interdependent network between power grid and communication network. The attacker has multiple attacking types. The defender needs a learning-based belief formation regarding attacking types, and then determine an equilibrium strategy

2.7 Conclusion

In this Chapter, we have aimed to illustrate that game theory can provide a sound mathematical approach to combat attacks across a wide range of applications. However, to do this one most go beyond the existing game theoretic models that typically assume additive utility functions, or that the attacker can attack only one target. While such assumptions have lead to tractable analyses, they miss key inherent dependencies that exist among different targets in current complex networks. In this chapter, we generalize the traditional security game model to the network scenario capturing network dependencies and the possibilities of a coordinated multi-resource attacks. We show that each security game is equivalent to a combinatorial optimization problem over a set system, which consists of defender's pure strategy space. The key technique we use is based on projection of a polytope based transformation, and the ellipsoid method. While in its most generality, capturing the equilibria under such an intricate model, is computationally hard, we provide several important classes of real-life problems for which our techniques can be used to develop optimal defense mechanisms. Based on our new mathematical framework, we outline a number of important future directions that can be investigated. The area of game theory coupled with reinforcement learning is fertile ground for solving many important security related problems.

Acknowledgement This work has been funded in part by a grant from the Army Research Office W911NF-15-1-0277.

References

1. R. Anderson, Why information security is hard - an economic perspective, in *Proceedings of ACSAC*, 2001
2. T. Moore, R. Anderson, Economics and internet security: a survey of recent analytical, empirical and behavioral research (2011). ftp://ftp.deas.harvard.edu/techreports/tr-03-11.pdf
3. T. Alpcan, T. Basar, *Network Security: A Decision and Game-Theoretic Approach* (Cambridge University Press, Cambridge, 2010)
4. L. Buttyan, J.P. Hubaux, *Security and Cooperation in Wireless Networks: Thwarting Malicious and Selfish Behavior in the Age of Ubiquitous Computing* (Cambridge University Press, Cambridge, 2007)
5. H. Kunreuther, G. Heal, Interdependent security. J. Risk Uncertain. **26**(2–3), 231–249 (2003)
6. M.H. Manshaei, Q. Zhu, T. Alpcan, T. Basar, Game theory meets network security and privacy. ACM Comput. Surv. **45**, 25 (2012)
7. A. Laszka, G. Horvath, M. Felegyhazi, L. Buttyan, Flipthem: modeling targeted attacks with flipit for multiple resources, in *Proceedings of GameSec*, 2014
8. M. Zhang, Z. Zheng, N.B. Shroff, A game theoretic model for defending against stealthy attacks with limited resources, in *GameSec 2015*, November 2015, London (Springer, Cham, 2015)
9. Z. Zheng, N.B. Shroff, P. Mohapatra, When to reset your keys: optimal timing of security updates via learning, in *AAAI'17*, San Francisco, CA, February 2017
10. J. Tsai, C. Kiekintveld, F. Ordonez, M. Tambe, S. Rathi, IRIS-a tool for strategic security allocation in transportation networks, in *Eighth International Joint Conference on Autonomous Agents and Multiagent Systems (Industry Track)*, May 2009
11. M. Brown, A. Sinha, A. Schlenker, M. Tambe, One size does not fit all: a game-theoretic approach for dynamically and effectively screening for threats, in *AAAI Conference on Artificial Intelligence (AAAI)*, 2016
12. F. Fang, A. Xin Jiang, M. Tambe, Optimal patrol strategy for protecting moving targets with multiple mobile resources, in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-Agent Systems* International Foundation for Autonomous Agents and Multiagent Systems (2013), pp. 957–964
13. M. Tambe, *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned* (Cambridge University Press, Cambridge, 2011)
14. A. Gueye, V. Marbukh, J.C. Walrand, Towards a metric for communication network vulnerability to attacks: a game theoretic approach, in *International Conference on Game Theory for Networks* (Springer, Berlin, 2012)
15. F.L. Sinong Wang, N.B. Shroff, Non-additive security games, in *AAAI*, 2017
16. S. Wang, N. Shroff, Security game with non-additive utilities and multiple attacker resources, in *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, vol. 1 (2017), p. 13
17. H. Von Stackelberg, *Marktform und gleichgewicht* (J. Springer, Berlin, 1934)

18. B. Von Stengel, S. Zamir, Leadership with commitment to mixed strategies, vol. 38. Technical report LSE-CDAM-2004-01, CDAM research report (2004)
19. F.V. Fomin, D. Kratsch, *Exact Exponential Algorithms* (Springer, Berlin, 2010)
20. R. Kennes, P. Smets, Computational aspects of the Mobius transformation, in *Proceedings of the Sixth Annual Conference on Uncertainty in Artificial Intelligence*, pp. 401–416 (Elsevier Science Inc., Amsterdam, 1990)
21. M. Grotschel, L. Lovasz, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization. Combinatorica **1**(2), 169–197 (1981)
22. R.J. Lipton, E. Markakis, A. Mehta, Playing large games using simple strategies, in *Proceedings of the 4th ACM Conference on Electronic Commerce (EC)*, pp. 36–41 (ACM, New York, 2003)
23. P. Shakarian, H. Lei, R. Lindelauf, Power grid defense against malicious cascading failure, in *Proceedings of the 2014 International Conference on Autonomous Agents and Multi-Agent Systems*. International Foundation for Autonomous Agents and Multiagent Systems (2014)
24. J. Pita, M. Jain, J. Marecki, F. Ordonez, 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 7th International Joint Conference on Autonomous Agents and Multiagent Systems: Industrial Track*, pp. 125–132. International Foundation for Autonomous Agents and Multiagent Systems (2013)
25. D. Korzhyk, V. Conitzer, R. Parr, Security games with multiple attacker resources, in *IJCAI Proceedings - International Joint Conference on Artificial Intelligence*, vol. 22 (2011), pp. 273–279. Citeseer
26. D. Korzhyk, V. Conitzer, R. Parr, Complexity of computing optimal Stackelberg strategies in security resource allocation games, in *AAAI*, 2010
27. F. Fang, T.H. Nguyen, R. Pickles, W.Y. Lam, G.R. Clements, B. An, A. Singh, M. Tambe, A. Lemieu, Deploying PAWS: field optimization of the protection assistant for wildlife security, in *Proceedings of the Twenty-Eighth Innovative Applications of Artificial Intelligence Conference*, 2016
28. H. Xu, F. Fang, A.X. Jiang, V. Conitzer, S. Dughmi, M. Tambe, Solving zero-sum security games in discretized spatio-temporal domains, in *AAAI* (2014), pp. 1500–1506, Citeseer
29. S.V. Buldyrev et al., Catastrophic cascade of failures in interdependent networks. Nature **464**(7291), 1025 (2010)

Chapter 3
Entropy-Based Proactive and Reactive Cyber-Physical Security

Aris Kanellopoulos and Kyriakos G. Vamvoudakis

Abstract This chapter considers the problem of securely operating a cyber-physical system under different types of attacks, including actuator and sensor attacks. The proposed defense approach consists of a proactive and a reactive mechanism. The proactive part leverages the principles of moving target defense, and introduces a stochastic switching structure that dynamically and continuously alters the behavior of the system, aiming to neutralize the attacker's reconnaissance efforts. An unpredictability metric is proposed that utilizes the entropy induced by a switching supervisor, in order to maximize efficiency. The reactive part isolates the potentially compromised system components. A novel integral Bellman-based intrusion detection system is used to detect the attacks and take appropriate measures by collecting data online and without knowledge of the physical interpretation of the system. Simulation results are presented to showcase the efficacy of the proposed approach.

3.1 Introduction

Cyber-physical systems (CPS) are complex platforms that consist of a multitude of heterogeneous physical components, integrated through communication protocols and operated via the appropriate software which implements the decision making algorithms [1]. CPS have found application in a variety of technological areas ranging from autonomous vehicles [2], medical applications [3], and smart grids [4, 5], to military operations. Similar to computer systems, CPS are prone to component vulnerabilities that pose a threat by leading to malicious exploits. However, the tight

A. Kanellopoulos · K. G. Vamvoudakis (✉)
The Guggenheim School of Aerospace Engineering, Georgia Institute of Technology, Atlanta, GA, USA
e-mail: ariskan@gatech.edu; kyriakos@gatech.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_3

interconnection of the physical and the information layers, results in an increase of the complexity of the system, and of the number of attack angles available to an adversarial agent.

Attacks on systems that leverage the coupling between the physical devices and the underlying software have been observed in numerous occasions [6]. Perhaps the most infamous real world CPS attack was due to the development of the Stuxnet virus, a computer worm that targeted programmable logic controllers [7]. Autonomous vehicles in particular have been shown to be vulnerable to malicious behaviors [2].

The need for deception stems from the inherent asymmetry created between the attacker and the defender in network security scenarios. On the one hand, the defender has to take into account all the subsystems of the network, as well as every conceivable exploit and fortify the network adequately. However, this task is extremely difficult, if not impossible when were one considers the existence of zero-day exploits. On the contrary, the attacker can be successful in compromising the system by discovering even a single vulnerability. Moreover, it is known that one of the most important, as well as time-consuming, phases of a cyber-attack takes place during system reconnaissance [8]. Thus, moving target defense (MTD) is introduced as a defense paradigm that aims to deceive the attacker by persistently changing the behavior of the system in an unpredictable way so as to invalidate the attacker's perception.

In this chapter, a novel algorithm that combines proactive and reactive security will be proposed. Instead of attempting to make the system resilient to attacks by sacrificing usability and optimality, we will assume that a continuously switching architecture equipped with an intrusion detection system will serve three distinct roles. Namely, (i) it will shrink the window of opportunity of the attacker by increasing the cost of estimating the exploits of the system; (ii) even under successful attacks, the system will be able to return to its healthy state; and (iii) by keeping compromised subsets of the system offline, the defender will be able to mitigate the damage that the system has suffered and trace the attacker.

While in this chapter we will focus on proactive defense that employs actuator and sensor redundancy, the present approach can be extended to leverage the increasing autonomous capabilities of CPS. The extension of block-chain and cloud services integrated with CPS only expands the possibilities of formulating a continuously shifting and impenetrable system of systems [9].

3.1.1 Related Work

The need to develop a security approach for CPS that examines the network as a whole, rather than utilize algorithms that operate only on the computational layer, was highlighted in [10]. Decision theory [11] offers mathematically rigorous tools for behavior analysis, even for large-scale and complex systems. Consequently, techniques have been proposed for security algorithm design through decision-theoretic concepts [12–14]. Among the different design approaches, optimal decision theory and game theory [15] have emerged as important frameworks due to their abilities to satisfy user-defined performances in adversarial environments.

In a more general game-theoretic context, in [11], network security problems are formulated as zero-sum games between the operators of the system and the attacking parties. Game theory is brought together with network security in a dynamical system setting in [16], where the authors introduce graphical game Nash equilibria and design optimal policies for interconnected dynamic agents under attacks. However, those approaches still allow the attacker to take advantage of the asymmetric nature of network security, since they seek to mitigate, rather than deter them. The authors in [17], employ trust metrics between the agents of a network.

Solutions to network security that leverage unpredictability and deception have been examined before, but they remained in the domain of the information layer [18–20]. More specifically, the authors in [21] utilize a constantly rotating Internet Protocol version 6 (IPv6) addressing to increase the network's privacy and anonymity. In [20], the authors introduce the Open-flow random host mutation scheme that assigns virtual, unpredictably changing, IPs to the network's hosts. A more formalized approach to MTD was introduced in [22], leading to an MTD entropy hypothesis framework that is generally applicable. The authors in [23] investigate the idea of MTD through a multilayer zero-sum game, where the action space of the defender consists of the different system configurations available. The solution to this game provides a stochastic policy that randomizes among those configurations in a dynamic fashion. From a control-theoretic perspective, an MTD approach has been used to enlarge the dimension of the state space in [24].

The authors in [25] fuse all the available sensor information, even if they are corrupted, and formulate a procedure for robust estimation if less than half of the sensors are compromised. The concept of redundancy through switching mechanisms, in order to relax such an assumption, has been leveraged in previous works. In [26, 27], the intrusion detection subsystem explicitly estimated the attack input, and switched the system via the principles of passivity-based control.

3.1.2 Structure

The remainder of the paper is structured as follows. Section 3.2, formulates the problem of securing a CPS from actuator and sensor attacks while also increasing the attacking surface to enhance uncertainty and unpredictability. In Sect. 3.3, we focus on proactive and reactive defense against actuator attacks. Section 3.4, extends the framework of Sect. 3.3, to incorporate a proactive and reactive defense framework against sensor attacks. Simulation results are shown in Sect. 3.5 and a discussion regarding the proposed algorithm are discussed in Sect. 3.6. Finally, Sect. 3.7 concludes and discusses future work.

3.1.3 Notation

The notation used in this paper is standard. $\bar{\lambda}(A)$ is the maximum eigenvalue of the matrix A and $\underline{\lambda}(A)$ is its minimum eigenvalue. $\|\cdot\|$ denotes the Euclidean norm of a vector and the Frobenius norm of a matrix. The superscript $\star$ is used to denote the optimal trajectories of a variable. $(\cdot)^{\mathrm{T}}$ denotes the transpose of a matrix. ∇_x and $\frac{\partial}{\partial x}$ are used interchangeably and denote the partial derivative with respect to a vector x. The cardinality of a set, i.e., the number of elements contained in the set, is denoted by card$(\cdot)$. 2^A denotes the power set of a set A, i.e., the set containing all the subsets of A, including the empty set and the A itself. Finally, supp(x) denotes the support of a vector, i.e., the number of its non-zero elements.

3.2 Problem Formulation

Consider the following nonlinear continuous-time system describing physical and cyber inter-dependencies,

$$\begin{aligned} \dot{x}(t) &= f(x) + g(x)u_a(t),\ t \geq 0, \\ y(t) &= h_a(x,t), \end{aligned} \tag{3.1}$$

where $x(t) \in \mathbb{R}^n$ is the state, $u_a(t) \in \mathbb{R}^m$ is the potentially attacked input of the system (decision policy), $y(t) \in \mathbb{R}^p$ is the output, $f(x) : \mathbb{R}^n \to \mathbb{R}$ are the drift dynamics, which show the evolution of the system's state in the absence of input, $g(x) : \mathbb{R}^n \to \mathbb{R}^{n\times m}$ are the input dynamics, which couple the input decision policy with the system's evolution, and $h_a(x,t) : \mathbb{R}^n \times \mathbb{R}^+ \to \mathbb{R}^p$ is the potentially attacked output function, that provides the measurable data in which the operator has access.

We can rewrite (3.1) as,

$$\begin{aligned} \dot{x}(t) &= f(x) + \sum_{i=1}^{m} g_i(x)u_i, \\ y_j(t) &= h_{aj}(x,t), j \in \{1,\ldots,p\}, \end{aligned}$$

where $g_i(x)$ is a column vector corresponding to the i-th actuator, u_i is the value of the input signal associated with this actuator, and y_j is the output given by a specific sensor $h_{aj}(x,t)$.

The potentially compromised input of (3.1) will have the following form,

$$u_a(t) = \rho(t)u(t), \tag{3.2}$$

where $\rho(t) = \mathrm{diag}(\rho_{ii}(t))$, $\forall i \in \{1, \ldots, m\}$ is a time varying actuator attack parameter controlled by an adversary and $u(t) \in \mathbb{R}^m$ is the non-attacked input.

The output function of the system can be undermined by a signal $\rho^s(t)$ as,

$$h_a(x,t) = \rho^s(t)h(x), \tag{3.3}$$

where $\rho^s(t)$ is a diagonal matrix controlled by the attacker and $h(x) : \mathbb{R}^n \to \mathbb{R}^p$ is the non-attacked output function.

Assumption 3.1 In order to offer a greater degree of freedom for deception purposes and to mitigate the effect of potential attacks, we will consider systems with redundant actuating and sensing components.

□

Assumption 3.2 We will assume that the system's actuators are not compromised over a time interval $\tau \in [t_1, t_2]$ if and only if $\rho_{ii}(t) = 1, \forall i \in \{1, \ldots, m\}, \forall \tau$. Similarly, we consider the sensors as secure, if and only if $\rho^s_{jj}(t) = 1, \forall j \in \{1, \ldots, p\}\ \forall \tau$. The signals (3.2) and (3.3) are assumed to be locally integrable over any closed time interval $[t_1, t_2]$, $0 \le t_1 < t_2$. □

Remark 3.1 It should be noted that our formulation will make no assumptions on the structure, on boundedness and other Lipschitz continuity properties of the attacker's signal. Furthermore, attacks of the form (3.2) and (3.3), due to their time-varying nature, can describe a wide range of attacks, including additive and multiplicative attacks.

□

Assumption 3.3 We will assume that the attacker is not able to compromise all of the actuators and sensors at once. Therefore, $\mathrm{supp}(\rho) < m$ and $\mathrm{supp}(\rho^s) < p$.

□

We are thus interested in designing a proactive and a reactive defense mechanism that will operate well in the absence of attackers, while also detecting and mitigating real-time attacks.

In order to derive closed-form solutions for the proactive decision and mitigation policies, as well as for the intrusion detection system, we will henceforth utilize a linearized form of the system (3.1) as,

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu_a(t),\ t \ge 0, \\ y(t) &= C_a(t)x(t), \end{aligned} \tag{3.4}$$

where $A \in \mathbb{R}^{n\times n}$ is the plant matrix, $B \in \mathbb{R}^{n\times m}$ is the input matrix, and $C_a(t) \in \mathbb{R}^{p\times n}$ is the potentially attacked output matrix.

Accordingly, the actuation and sensing redundancy can be highlighted by expressing the system in the form,

$$\begin{aligned} \dot{x} &= Ax + \sum_{i=1}^{m} b_i u_i, \\ y_j &= c_{aj}x, j \in \{1, \ldots, p\}, \end{aligned}$$

where b_i is a column vector corresponding to the i-th actuator, u_i is the value of the input signal associated with this actuator, and y_j is the output given by a specific sensor c_{aj} corresponding to the j-th row of the output matrix $C_a(t) = \rho^s(t)C$, where C is the output matrix in the absence of attacks.

3.3 Defense Against Actuator Attacks

We will initially focus our attention to the case of actuator attacks. Let $\mathscr{B}$ denote the set containing the actuators of (3.4) by the vectors b_i, $i \in \{1, \ldots, m\}$. The power set of $\mathscr{B}$, denoted as $2^{\mathscr{B}}$, contains all possible combinations of the actuators acting on (3.4). Each of these combinations is expressed by the input matrix $B_j, j \in \{1, \ldots, m\}$ whose columns are the appropriate vectors b_i.

The set of the candidate actuating modes $\mathscr{B}_c$ is defined as the set of the actuator combinations that renders the system (3.4) fully controllable, i.e.,

$$\mathscr{B}_c = \left\{ B_j \in 2^{\mathscr{B}} : \text{rank}(\left[B_j \ \ AB_j \ \ldots \ A^{n-1}B_j\right]) = n \right\}. \tag{3.5}$$

The system (3.4) assuming full state-feedback, with the actuating mode B_i can be rewritten as,

$$\dot{x} = Ax + B_i u_i, \ i \in \{1, \ldots, m\}, \ t \geq 0. \tag{3.6}$$

Remark 3.2 Note that, we do not require different actuating modes to share common actuators. Moreover, while a single actuating mechanism might be able to drive and mitigate a system, two different—less potent—mechanisms might need to work cooperatively for the same system. All these modes will belong to the set described in (3.5).

□

3.3.1 Optimal Feedback Policies Design

For each actuating operating mode B_i, $i \in \{1, \ldots, m\}$, we denote the candidate policy as $u_i(t)$.

We are interested in deriving optimal decision policies for each of these modes. Towards that, we are interested in solving the following optimization,

$$\begin{aligned} V_i^\star(x(t_0)) &= \min_{u_i} \int_{t_0}^{\infty} r_i(x, u_i)\mathrm{d}\tau \\ &\equiv \min_{u_i} \int_{t_0}^{\infty} (x^{\mathrm{T}} Q_i x + u_i^{\mathrm{T}} R_i u_i)\mathrm{d}\tau, \ \forall x(t_0), \end{aligned} \tag{3.7}$$

given (3.6), where $Q_i \succeq 0$, $R_i \succ 0$, $\forall i \in \{1, \ldots, \text{card}(\mathscr{B}_c)\}$ Intuitively, the matrices Q_i penalize the deviation from the zero state, while the matrices R_i penalize the energy content of the feedback policies.

Assumption 3.4 We assume that each pair $(A, \sqrt{Q_i})$ is detectable.

□

The Hamiltonian associated with (3.6) and (3.7) is,

$$H_i(x, u_i, \nabla V_i) = \nabla V_i^{\mathrm{T}}(Ax + B_i u_i) + x^{\mathrm{T}} Q_i x + u_i^{\mathrm{T}} R_i u_i,\ \forall x, u_i,$$

with V_i denoting the value function, not necessarily the optimal.

Applying the stationarity conditions $\frac{\partial H_i(x,u_i,\nabla V_i)}{\partial u_i} = 0$, yields,

$$u_i = -R_i^{-1} B_i^{\mathrm{T}} \nabla V_i. \tag{3.8}$$

The optimal value functions $V_i^\star(\cdot)$ must satisfy the following HJB equation,

$$x^{\mathrm{T}} Q_i x + \nabla V_i^{\star \mathrm{T}} A x - \frac{1}{2} \nabla V_i^{\star \mathrm{T}} B_i R_i^{-1} B_i^{\mathrm{T}} \nabla V_i^\star = 0. \tag{3.9}$$

Since all the systems described by (3.6) are linear and the cost given by (3.7) is quadratic, all the value functions will be quadratic in the state x, i.e., $V_i^\star(x) = x^{\mathrm{T}} P_i x$. Substituting this expression into (3.9) and the resulting optimal value function into (3.8), yields the feedback policy with optimal gain K_i,

$$u_i^\star(x) = -K_i x := -R_i^{-1} B_i^{\mathrm{T}} P_i x,\ \forall x,$$

where $P_i \succ 0$ are the solutions to the following Riccati equations,

$$A^{\mathrm{T}} P_i + P_i A - P_i B_i^{\mathrm{T}} R_i^{-1} B_i^{\mathrm{T}} P_i + Q_i = 0. \tag{3.10}$$

We introduce $\mathscr{K}$ as the set containing all K_i, $i \in \{1, \ldots, m\}$, with the understanding that $\text{card}(\mathscr{K}) = \text{card}(\mathscr{B}_c)$. For ease of exposition, with some abuse of notation we will consider K_i to mean the optimal policy with this gain as well as its corresponding index.

Fact 3.1 Due to (3.5) and Assumption 3.4, for each B_i, the solution exists and is unique. □

Fact 3.2 Each K_i, with input given by (3.8) guarantees that (3.4) has an asymptotically stable equilibrium point. □

3.3.2 Switching-Based MTD Framework

We will now develop a framework to facilitate deception of potential attackers based on the principles of MTD. MTD algorithms aim to change the parameters of the network in an unpredictable and stochastic fashion therefore increasing the cost of attack, limiting the exposure of vulnerable components, and deceiving the opponent. The assumption of redundancy on the actuation and sensor systems is required to offer the intelligent secure policy the appropriate degree of freedom to alter parameters, conceal vulnerable components and isolate potentially compromised ones, without risking the safe operation of the network. The MTD Hypothesis states that MTD systems are more successful the more unpredictable and chaotic they are. To define an unpredictability metric to optimize, we will consider the entropy induced by the proposed dynamic network security algorithm.

3.3.2.1 Maximization of Unpredictability

To formally define the switching law, we need to introduce the *probability simplex* **p**, which denotes the probability that each feedback policy K_i is active.

To incorporate ideas from the framework of MTD, we propose a switching rule that optimizes over the minimum cost that each feedback policy is able to attain, as well as an unpredictability term quantified by the information entropy produced by the switching probability simplex **p**. This way, we will achieve the desired trade-off between overall optimality and unpredictability. The use of information entropy is a standard practice in MTD design [28].

Theorem 3.1 *Suppose that* (3.4)*, is driven by* $N = \text{card}(\mathscr{K})$ *candidate policies with an associated cost given by* (3.7)*. Then, the probability* p_i *that each policy* K_i *is active is given by*,

$$p_i = \mathrm{e}^{\left(-\frac{V_i}{\epsilon}-1-\epsilon\log\left(\mathrm{e}^{-1}\sum_{i=1}^{N}e^{\frac{V_i}{\epsilon}}\right)\right)}, \tag{3.11}$$

with $\epsilon \in \mathbb{R}^+$ *denoting the level of unpredictability.*

Proof We formulate the following optimization problem,

$$\min_{\mathbf{p}}\left(\mathbf{V}^{\star\mathrm{T}}\mathbf{p}-\epsilon\mathscr{H}(\mathbf{p})\right)$$

$$\text{subject to} : \|\mathbf{p}\|_1 = 1 \text{ and } \mathbf{p} \succeq 0,$$

where $\mathbf{V}^\star := \left[V_1^\star \dots V_N^\star\right]^{\mathrm{T}} = \left[x(t_0)^{\mathrm{T}}P_1x(t_0)\dots x(t_0)^{\mathrm{T}}P_Nx(t_0)\right]^{\mathrm{T}}$ denotes a column vector containing the value function of each candidate policy, $\mathscr{H}(\mathbf{p}) = -\mathbf{p}^{\mathrm{T}}log(\mathbf{p})$ is the information entropy produced by the simplex. Furthermore, for the decision vector $\mathbf{p}$ to constitute a probability simplex we constrain it to the non-negative orthant (i.e. $p_i \leq 0$, $\forall i \in \{1 \ldots N\}$) and we require its l_1 norm to satisfy, $\|p\|_1 = \sum_{i=1}^{N} \|p_i\| = 1$.

The entropy of a probability is a concave function [29] and therefore, the cost index, being a sum of a linear function of the probability and the negative entropy, is convex. Thus, we can define the Lagrangian of the optimization problem as,

$$\begin{aligned} L &= \mathbf{V}^{\star\mathrm{T}}\mathbf{p} - \epsilon\mathscr{H}(\mathbf{p}) + \lambda(\mathbf{1}^{\mathrm{T}}\mathbf{p} - 1) + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{p} \\ &= \mathbf{V}^{\star\mathrm{T}}\mathbf{p} + \epsilon\mathbf{p}^{\mathrm{T}}\log(\mathbf{p}) + \lambda(\mathbf{1}^{\mathrm{T}}\mathbf{p} - 1) + \boldsymbol{\beta}^{\mathrm{T}}\mathbf{p}, \end{aligned}$$

where $\mathbf{1}$ denotes a vector consisting of ones of appropriate dimensions, and λ, β are the Karush-Kuhn-Tucker (KKT) multipliers.

The KKT conditions for the problem are,

$$\nabla_{\mathbf{p}}L = \mathbf{V}^{\star} + \epsilon\mathbf{1} + \epsilon\log(\mathbf{p}) + \lambda\mathbf{1} + \boldsymbol{\beta},$$

and the complementarity conditions for the optimal solution $\mathbf{p}^{\star}$ are,

$$\boldsymbol{\beta}^{\mathrm{T}}\mathbf{p}^{\star} = 0.$$

If there exists an i for which $p_i = 0$, then the term $\log(p_i)$ will be undefined. Consequently, for the optimization problem to be feasible, one of the following two conditions need to hold,

- $\epsilon\log(p_i) = 0, \forall i \Rightarrow \epsilon = 0 \Rightarrow \mathbf{p}^{\star} = \left[\mathbf{0}_{i-1} \ldots 1 \ldots \mathbf{0}_{N-i}\right]^{\mathrm{T}}$ where the K_i feedback policy is the one with the overall less cost.
- $\boldsymbol{\beta} = 0$.

Consider now the nontrivial case, i.e., $\boldsymbol{\beta} = 0$, which yields,

$$\nabla_{\mathbf{p}}L = \mathbf{V}^{\star} + \epsilon\log(\mathbf{p}) + \epsilon\mathbf{1} + \lambda\mathbf{1} = 0.$$

The N equations for each policy are independent, leading to the following system of equations,

$$V_i^{\star} + \epsilon\log(p_i) + \epsilon + \lambda = 0, \forall i \in \{1, \ldots, N\}.$$

Solving now for the optimal probabilities p_i, yields,

$$p_i = \mathrm{e}^{\left(-\frac{V_i^{\star}}{\epsilon} - \frac{\lambda}{\epsilon} - 1\right)}, \forall i \in \{1, \ldots, N\}. \tag{3.12}$$

Taking into account that,

$$\|\mathbf{p}\|_1 = 1 \Rightarrow \sum_{i=1}^{N} p_i = 1 \Rightarrow \sum_{i=1}^{N} \mathrm{e}^{\left(-\frac{V_i^{\star}}{\epsilon} - \frac{\lambda}{\epsilon} - 1\right)} = 1.$$

and solving for λ, yields,

$$\lambda = \epsilon \log \left(\mathrm{e}^{-1} \sum_{i=1}^{N} \mathrm{e}^{(-\frac{V_i^\star}{\epsilon})} \right). \tag{3.13}$$

Substituting (3.13) in (3.12) provides the required result.

3.3.2.2 Switching-Based MTD Scheme

In order to formally define the behavior of the system under the proposed MTD framework, we shall formulate a switched system consisting of the different operating modes.

First, we introduce the switching signal $\sigma(t) = i$, $i \in \{1, \ldots, \mathrm{card}(\mathscr{K})\}$, which denotes the active policy as a function of time. This way, the system is,

$$\dot{x}(t) = \tilde{A}_{\sigma(t)} x(t),$$

where $\tilde{A}_{\sigma(t)} := A - B_{\sigma(t)} R^{-1}_{\sigma(t)} B^{\mathrm{T}}_{\sigma(t)} P_{\sigma(t)}$ denotes the closed-loop subsystem with the policy $K_{\sigma(t)}$ active.

Remark 3.3 Since the actual switching sequence is different under the designer's choice for unpredictability, we will constrain the switching signal to have a predefined average dwell time. This way, the stability of the overall system will be independent of the result of the optimization. Intuitively, as was initial shown in [30], a system with stable subsystems is stable if the switching is slow enough on an average sense. □

Theorem 3.2 *Consider the system* (3.4) *in the absence of attacks. The switched system defined by the piecewise continuous switching signal* $\sigma(t) = i$, $i \in \{1, \ldots, \mathrm{card}(\mathscr{K})\}$, *with active policy* K_i *given by* (3.8) *and continuous flow given by* (3.6) *has an asymptotically stable equilibrium point for every switching signal* $\sigma(t)$ *if the average dwell time is bounded by*,

$$\tau_D > \frac{\max_{q,p \in \{1,\ldots,\mathrm{card}(\mathscr{K})\}} \frac{\bar{\lambda}(P_p)}{\underline{\lambda}(P_q)}}{\min_{p \in \{1,\ldots,\mathrm{card}(\mathscr{K})\}} \frac{\underline{\lambda}(Q_p + P_p B_p R_p^{-1} B_p^{\mathrm{T}} P_p)}{\underline{\lambda}(P_p)}},$$

Proof The proof is given in [31].

3.3.3 Integral Bellman-Based Intrusion Detection Mechanism

In this subsection, an intrusion detection mechanism is designed to identify potentially corrupted sets of feedback policies that belong in the set $\mathscr{K}$. The attack detection signal will rely on the optimality property as well as on data measured along the—possibly corrupted—trajectories of the system. Based on a sampling mechanism, we denote the measurements of the state at the sampling instances as $x_c(t)$ and define the functions $\hat{V}_i(\cdot) := x_c^{\mathrm{T}} P_i x_c$, $i \in \{1, \dots, \mathrm{card}(\mathscr{K})\}$. Intuitively, we obtain a sampled version of the optimal value function along the system's real, and potentially compromised, trajectories.

Theorem 3.3 *Consider that the system is operating with $K_i \in \mathscr{K}$, designed based on* (3.8) *and* (3.9). *Define the detection signal over a predefined time window $T > 0$,*

$$\begin{aligned} e(t) = \hat{V}_i(x_c(t-T)) - \hat{V}_i(x_c(t)) \\ - \int_{t-T}^{t} (x_c^{\mathrm{T}} Q_i x_c + u_i^{\star\mathrm{T}} R_i u_i^{\star}) \mathrm{d}\tau. \end{aligned} \tag{3.14}$$

Then, the system is under attack if and only if $e(t) \neq 0$. The optimality loss due to the attacks, quantified by $\|e(t)\|$, is bounded for any injected signal $\rho(t)$ that is integrable.

Proof As was proven in [32], Eq. (3.14) is the integral form of the Bellman equation. For the sampled value of the state at $t_1 = t - T$, we have that,

$$\begin{aligned} \hat{V}_i(t-T) &= x^{\mathrm{T}}(t-T) P_i x(t-T) \\ &= \min_{u_i} \Big\{ \int_{t-T}^{t} (x^{\mathrm{T}} Q_i x + u_i^{\mathrm{T}} R_i u_i) \mathrm{d}\tau + \hat{V}_i(t) \Big\}. \end{aligned}$$

Since $P_i \succ 0$ we have,

$$\begin{aligned} \hat{V}_i(t-T) &= x^{\mathrm{T}}(t-T) P_i x(t-T) \\ &= \min_{u_i} \Big\{ \int_{t-T}^{t} (x^{\mathrm{T}} Q_i x + u_i^{\mathrm{T}} R_i u_i) \mathrm{d}\tau \Big\} + x^{\mathrm{T}}(t) P_i x(t). \end{aligned}$$

For the accumulated cost utilizing the optimal input, and the cost utilizing an arbitrary input u_a, it is true that,

$$\begin{aligned} \int_{t-T}^{t} (x^{\mathrm{T}} Q_i x + u_i^{\star} R_i u_i^{\star}) \mathrm{d}\tau &= \min_{u_i} \Big\{ \int_{t-T}^{t} (x^{\mathrm{T}} Q_i x + u_i^{\mathrm{T}} R_i u_i) \mathrm{d}\tau \Big\} \\ &\leq \int_{t-T}^{t} (x^{\mathrm{T}} Q_i x + u_a^{\mathrm{T}} R_i u_a) \mathrm{d}\tau \Rightarrow \\ \int_{t-T}^{t} (x^{\mathrm{T}} Q_i x + u_i^{\star} R_i u_i^{\star}) \mathrm{d}\tau &= \int_{t-T}^{t} (x^{\mathrm{T}} Q_i x + u_a^{\mathrm{T}} R_i u_a) \mathrm{d}\tau - I(\rho). \end{aligned}$$

Due to Assumption 3.4, the solution is unique. By extension, the optimal cost over any time interval is also unique. Consequently, the system is not under attack only when $I(\rho) = 0$.

Algorithm 1: Proactive/reactive defense mechanism for actuator attacks

```
 1: procedure
 2:     Given initial state $x(t_0)$, and time window $T$.
 3:     Find all permutations of actuators (columns of $B$) and derive the subset of controllable
    pairs $(A, B_i)$, denoted by $\mathscr{K}$.
 4:     for $i = 1, \dots, \mathrm{card}(\mathscr{K})$
 5:         Compute the optimal feedback gain and Riccati matrices $K_i$, $P_i$ according to (3.8) and
    (3.10).
 6:         Compute the optimal cost of each feedback policy for the given $x(t_0)$.
 7:     end for
 8:     Solve for the optimal probabilities $p_i^\star$ using (3.11).
 9:     At $t = t_0$, choose the optimal policy for which $\sigma(t_0) = \arg\min_i \big(x(t_0)^\mathrm{T} P_i x(t_0)\big)$.
10:     while $\sigma(t) = i$ and $t < \tau_D$
11:         Compute the integral Bellman error detection signal using (3.14).
12:         Propagate the system using (3.6).
13:     end while
14:     Choose the random mode $\sigma(t + \tau_D) = j$ and go to 9.
15:     if $\|e_i(t_c)\| > 0$
16:         Take the $i$-th feedback policy implementation offline.
17:         Switch to the policy with the best performance,

            $$\sigma(t_c) = \arg \min_{i \in \mathscr{K} \setminus i} \big(x(t_0)^\mathrm{T} P_i x(t_0)\big)$$

    and go to 9.
18:     end if
19: end procedure
```

Fact 3.3 It has been shown, that $p_i > 0, \forall i \in \{1, \dots, \mathrm{card}(\mathscr{K})\}$. Consequently, there exists a $t_f^\star$ such that there $\exists \tau \in [t_0, t_f^\star]$ with $\sigma(\tau) = i \quad \forall i \in \{1, \dots, \mathrm{card}(\mathscr{K})\}$ and an arbitrary $t_0 > 0$. This means that, since the probability that all policies will eventually be active, there is some time interval long enough, such that we have already switched through every available policy. This way we can guarantee that eventually every compromised policy will have become active and subsequently isolated. □

Theorem 3.4 *Suppose that the* (3.4)*, uses the framework of Algorithm* 1*. Then the closed-loop system has an asymptotically stable equilibrium point given that the attacker has not compromised all the available feedback policies, i.e.,* $\mathscr{K} \setminus \mathscr{K}_c \neq \emptyset$*, where* $\mathscr{K}_c$ *is the subset of those policies that have been compromised by an attacker.*

3.3.3.1 Intrusion Detection Under Actuation Noise

It is possible to extend the results of the previous section to take into account noise in the actuation mechanism, i.e. in (3.4),

$$u_a(t) = \rho(t)u^\star(t) + w(t),$$

where $w(t)$ is a bounded but otherwise unknown disturbance with $\|w(t)\| \leq \bar{w}$.

Theorem 3.5 *The system* (3.4)*, equipped with the MTD decision scheme described in Sect.*3.3 *and the detection mechanism as defined in Theorem* 3.3*, under the effect of a disturbance $w(t)$ is compromised if,*

$$\|e(t)\| \geq e_{i,\text{thres}}(t)$$

where $e_{i,thres}$ are dynamic thresholds for each feedback mode of the form,

$$e_{i,thres}(t) = 2\|\bar{w}\| \int_{t-T}^{t} \|R_i u_i^\star(\tau)\| \mathrm{d}\tau + \bar{\lambda}(R_i)\|\bar{w}\|^2.$$

Proof First, we will consider the system in the absence of attacks and formulate the intrusion detection signal based on the data along the trajectories of the system. In other words we can write,

$$\begin{aligned}
e(t) &= \hat{V}_i(t-T) - \hat{V}_i(t) - \int_{t-T}^{t} (x^\mathrm{T} Q_i x + u_a^\mathrm{T} R_i u_a)\mathrm{d}\tau \\
&= \hat{V}_i(t-T) - \hat{V}_i(t) - \int_{t-T}^{t} \big(x^\mathrm{T} Q_i x + (u_i^\star + w)^\mathrm{T} R_i (u_i^\star + w)\big)\mathrm{d}\tau \\
&= \hat{V}_i(t-T) - \hat{V}_i(t) - \int_{t-T}^{t} (x^\mathrm{T} Q_i x + u_i^{\star\mathrm{T}} R_i u_i^\star)\mathrm{d}\tau \\
&\quad - \int_{t-T}^{t} (w^\mathrm{T} R_i u^\star + u_i^{\star\mathrm{T}} R_i w + w^\mathrm{T} R_i w)\mathrm{d}\tau.
\end{aligned}$$

Leveraging the integral Bellman equality and taking norms, yields,

$$\begin{aligned}
&\|e(t)\| \leq 2\int_{t-T}^{t} \|w^\mathrm{T} R_i u_i^\star\|\mathrm{d}\tau + \int_{t-T}^{t} \|w^\mathrm{T} R_i w\|\mathrm{d}\tau \Rightarrow \\
&\|e(t)\| \leq 2\|\bar{w}\| \int_{t-T}^{t} \|R_i u_i^\star(\tau)\|\mathrm{d}\tau + \bar{\lambda}(R_i)\|\bar{w}\|^2,
\end{aligned}$$

which is the adaptive threshold for the active feedback i.

Remark 3.4 It should be noted that the adaptive threshold can be computed online utilizing only knowledge of the optimal input signal that the feedback policy sends to the system (and not the potentially corrupted one). □

3.4 Defense Against Sensor Attacks

In this section we show how the methods developed can be applied to securely estimating the state of a system with compromised measurements by employing sensor redundancy.

3.4.1 Candidate Sensors Sets

Similarly to the proposed framework for the actuators, we introduce the set of all sensors, denoted by $\mathscr{C}$, and the elements of its power set $\mathscr{C}_i \in 2^{\mathscr{C}}$, where $C_i \in \mathscr{C}_i$ is a combination of the different rows of C.

The set of candidate sensing modes $\mathscr{S}_o$ is defined as the set of the sensor combinations that renders the system (3.4) fully observable,

$$\mathscr{S}_o = \left\{C_j \in 2^{\mathscr{C}} : \text{rank}\left(\begin{bmatrix} C_j \\ C_j A \\ \vdots \\ C_j A^{n-1} \end{bmatrix}\right) = n\right\}.$$

The system utilizing the sensor combination C_i is,

$$\begin{aligned} \dot{x} &= Ax + Bu, \\ y_i &= C_i x. \end{aligned}$$

Remark 3.5 We note the distinction between the set of sensors $\mathscr{C}$ and the set of sensing modes $\mathscr{S}_o$. The set of sensors contains the different physical components that measure parts of the system's behavior. On the other hand, the set of sensing modes contains those cooperating sensors together with an observer scheme that reconstruct an estimate of the system state. □

3.4.2 Optimal Observer Design

The observer of (3.4) will be now designed as a dynamic system sharing the same structural properties,

$$\begin{aligned} \dot{\hat{x}} &= A\hat{x} + Bu + B\bar{u}_i, \\ \hat{y}_i &= C_i \hat{x}, \end{aligned} \tag{3.15}$$

where $\hat{x}$, $\hat{y}_i$ are the estimates of the state and the output respectively, $\bar{u}_i$ denotes a "fictional" input, i.e., a correction term which forces the observer to track the actual system.

Remark 3.6 The state estimate $\hat{x}$ is independent of the active sensing mode. On the other hand, the output $\hat{y}_i$ and "fictional" input $\bar{u}_i$ are not. □

To design the optimal $\bar{u}_i$, we define the optimization problem based on the following cost function $\forall t \geq 0$,

$$U_i^\star(\hat{x}) = \min_{\bar{u}_i} \int_t^\infty \Big[(\hat{y}_i - y_i)^{\mathrm{T}} Q_i (\hat{y}_i - y_i) + \bar{u}_i^{\mathrm{T}} R_i \bar{u}_i\Big] \mathrm{d}\tau.$$

Defining the Hamiltonian of the system as,

$$\begin{aligned} H_i(\hat{x}, \bar{u}_i^\star, U_i^\star) = (\hat{y}_i - y_i)^{\mathrm{T}} Q_i (\hat{y}_i - y_i) + \bar{u}_i^{\star\mathrm{T}} R_i \bar{u}_i^\star \\ + \nabla U_i^{\star\mathrm{T}} (A\hat{x} + Bu + B\bar{u}_i^\star) = 0. \end{aligned} \tag{3.16}$$

We can now find the optimal policy from the stationarity conditions $\frac{\partial H_i(\hat{x}, \bar{u}_i^\star, U_i^\star)}{\partial \bar{u}_i^\star} = 0$. This leads to,

$$\bar{u}_i^\star = -R_i^{-1} B^{\mathrm{T}} \nabla U_i^\star(\hat{x}).$$

Due to the quadratic structure of the cost functional and the linear structure of the dynamic system, we assume that the value function is quadratic in $\hat{x}(t)$, i.e., $U_i^\star(\hat{x}) = \hat{x}^{\mathrm{T}} G_i \hat{x}$, which means that the optimal 'input' is,

$$\bar{u}_i^\star = -R_i^{-1} B^{\mathrm{T}} G_i \hat{x}. \tag{3.17}$$

In this section, we show how the same techniques introduced and analyzed in the previous sections can be applied to detect and mitigate sensor attacks.

3.4.3 MTD for Sensor Attacks

Theorem 3.6 *The state estimation scheme utilizing optimal observers as described by* (3.15)*, for every sensing mode in $\mathscr{S}_o$ has an asymptotically stable equilibrium point under a switching-based MTD mechanism given that the switching signal has the average dwell time,*

$$\tau_D > \frac{\max_{q,p\in\{1,\ldots,\mathrm{card}(\mathscr{S}_o)\}} \frac{\bar{\lambda}(G_p)}{\underline{\lambda}(G_q)}}{\min_{p\in\{1,\ldots,\mathrm{card}(\mathscr{S}_o)\}} \frac{\underline{\lambda}(C_i^{\mathrm{T}} Q_p C_i + G_p B_p R_p^{-1} B_p^{\mathrm{T}} G_p)}{\underline{\lambda}(G_p)}}.$$

Proof The proof is given in [31].

Remark 3.7 The optimization problem solved in Sect. 3.3 is identical for the case of sensor switching. As a result, the probability that a certain sensing mode S_i is active, obeys (3.11). The ideas from Sect. 3.3 directly apply to the sensor problem. □

3.4.4 Integral Bellman Based Intrusion Detection for Sensor Attacks

We will now introduce a detection signal based on the online, possibly compromised, estimations of the state, which we will denote $\hat{x}_c(t)$. For that reason, we formulate the function $\hat{U}_i(t) = \hat{x}_c^{\mathrm{T}} G_i \hat{x}_c$.

Theorem 3.7 *Consider the system* (3.4) *operating with the sensor mode* $S_i \in \mathscr{S}$*, designed based on* (3.16) *and* (3.17)*. Define the detection signal over a predefined time window* $T > 0$ *as,*

$$e^s(t) = \hat{U}_i(\hat{x}_c(t-T)) - \hat{U}_i(\hat{x}_c(t)) - \int_{t-T}^{t} \big((y_i - \hat{y}_i)^{\mathrm{T}} Q_i (y_i - \hat{y}_i) + \bar{u}_i^{\star\mathrm{T}} R_i \bar{u}_i^{\star}\big)\mathrm{d}\tau.$$

Then, the system is under attack if and only if $e^s(t) \neq 0$*. Moreover, the optimality loss due to attacks, is bounded for any injected signal* $\rho^s(t)$*.*

3.4.5 Proactive and Reactive Defense for Sensor Attacks

We will now combine the proactive defense mechanism with the intrusion detection system described above. The pseudo-code for the operation is presented in Algorithm 2.

Remark 3.8 We can combine the algorithmic frameworks presented for both actuators and sensor attacks. However, the result would be conservative, since the two problems are coupled. Consequently, we cannot differentiate between integral Bellman errors caused by an actuator or a sensor attack. □

Algorithm 2: Proactive/reactive defense mechanism for sensor attacks

```
1:  procedure
2:      Given initial state $x(t_0)$, system dynamics (3.4) and time window $T$.
3:      Find all permutations of sensors (rows of $C$) and derive the subset of observable pairs
    $(A, C_i)$, denoted $\mathscr{S}_o$.
4:      for $i = 1, \ldots, \mathrm{card}(\mathscr{S}_o)$
5:          Compute the optimal ‘fictional’ input and value function according to (3.16) and (3.17).
6:          Compute the optimal cost of each observation mode for the given $x(t_0)$.
7:      end for
8:      Solve for the optimal probabilities $p_i^\star$ using (3.11).
9:      At $t = t_0$, choose the optimal observer.
10:     while $\sigma(t) = i$ and $t < \tau_D$
11:         Compute the integral Bellman error detection signal using (3.18).
12:         Propagate the system using the observer dynamics.
13:     end while
14:     Choose a random mode $\sigma(t + \tau_D) = j$ and go to 9.
15:     if $\|e^s(t_c)\| > 0$
16:         Take the $i$-th observer offline.
17:         Switch to the safe observer with the best performance and go to 9.
18:     end if
19: end procedure
```

3.5 Simulation Results

In this section, we present simulation results to showcase the operation, efficacy, and shortcomings of the proposed approach.

Since redundancy is standard practice in the defense and aviation industries, where safety factors often require two to four redundant systems, it is natural to showcase the efficacy of our results in intelligent transportation systems. Towards this, we utilize the linearized model of a submarine as described in [33] as well as the benchmark ADMIRE aircraft [34]. Those systems are equipped with redundant actuators and sensors, allowing both proactive switching as well as enforced isolation of the compromised components.

Figure 3.1, shows the states of the underwater vehicle under the MTD switching mechanism. System operation above the minimum switching time threshold, guarantees that the system is stable. Figure 3.2 shows the state of the system under attack while the combined proactive/reactive framework is operational and Fig. 3.3 the switching signal. The actuating mode under attack corresponds to switching signal $i = 3$, which is also the most optimal. We notice that after the attack is detected, that mode is taken out of the switching queue.

Next, we highlight the use of the proposed intrusion detection algorithm for sensor attacks. The objective is to correctly estimate the setpoint of the angle of attack for the ADMIRE aircraft. Notice that we take into account the presence of sensor noise, and derive an adaptive intrusion threshold similar to the one examined in Theorem 3.5. Figure 3.4 shows the true and estimated values of the angle of attack, while Fig. 3.5, the integral Bellman error.

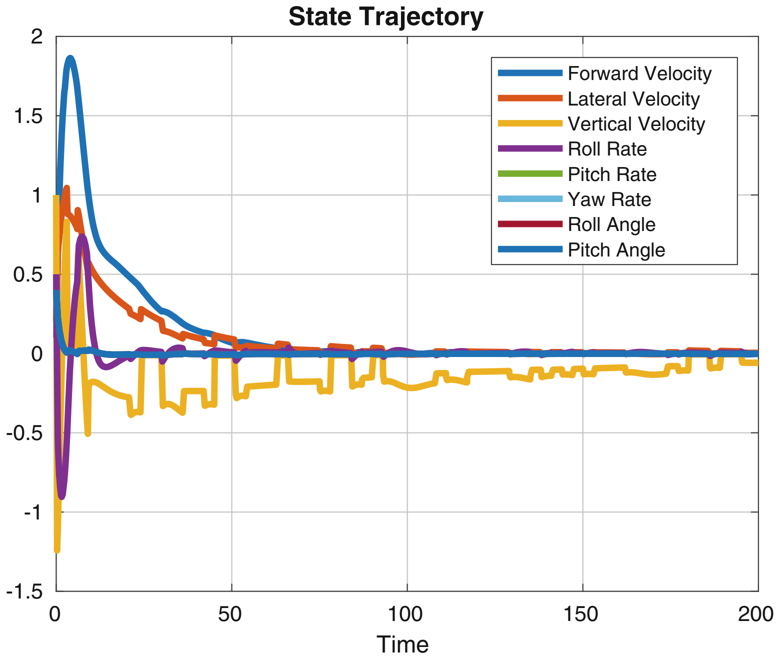


Fig. 3.1 Aircraft states under the persistent proactive actuator switching

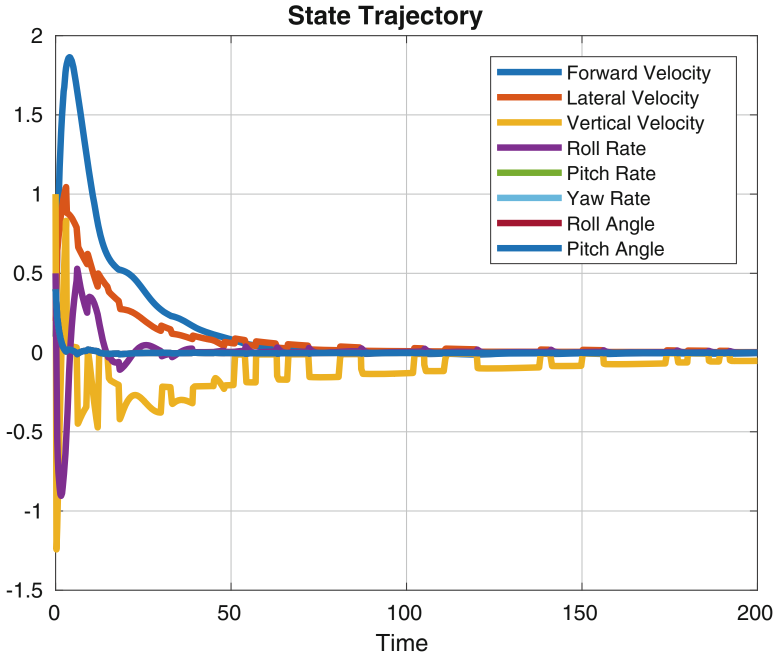


Fig. 3.2 The evolution of the state with both proactive and reactive defense

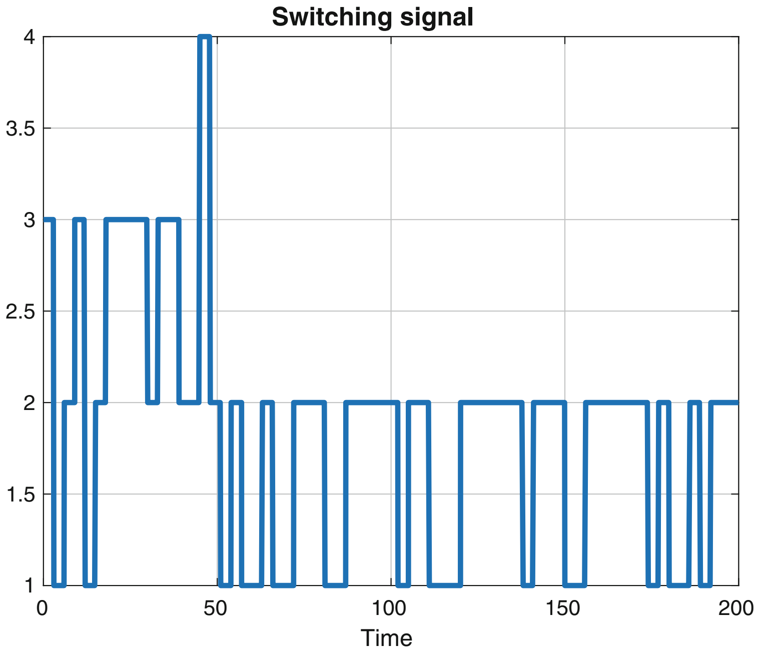


Fig. 3.3 The evolution of the switching signal with both proactive and reactive defense

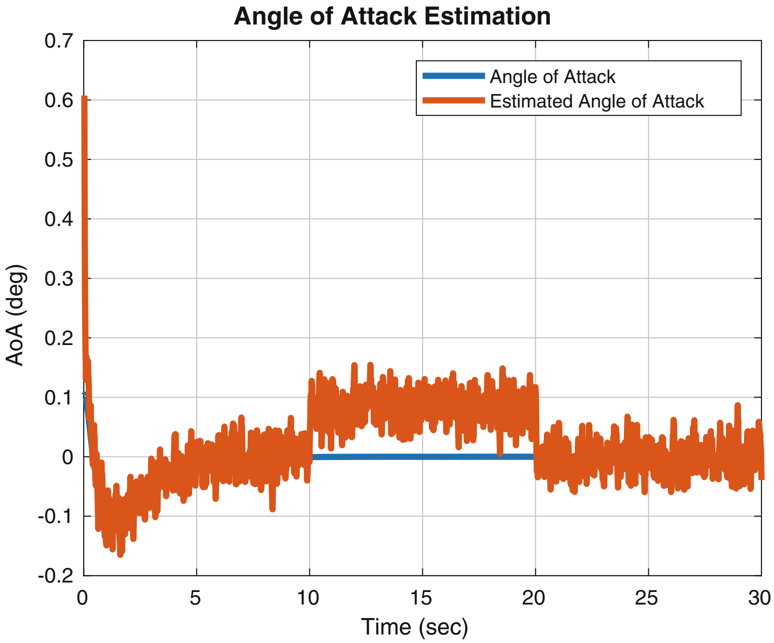


Fig. 3.4 True and estimated angle of attack under sensor noise and malicious input

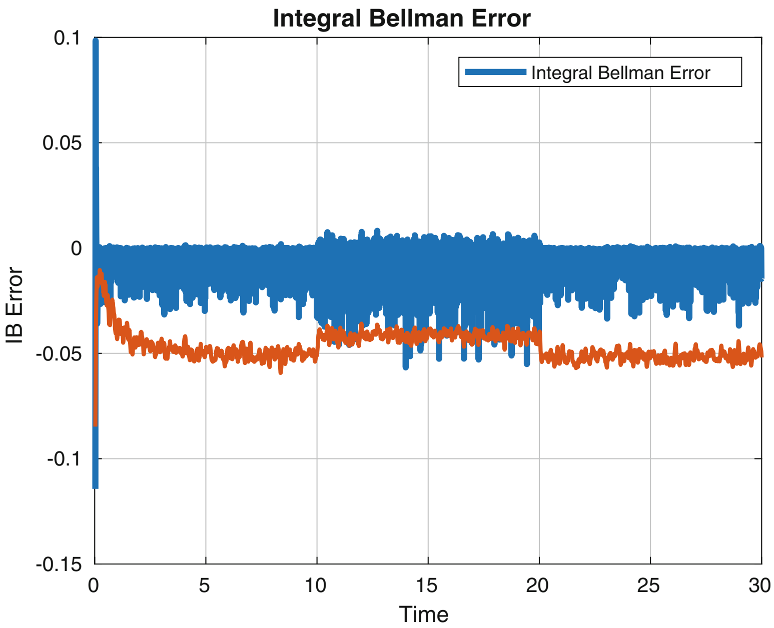


Fig. 3.5 The evolution of the integral Bellman error and adaptive threshold in the presence of sensor attacks

3.6 Discussion

One of the major advantages that the proposed integral Bellman-based intrusion detection system offers, is that it can be extended to systems whose dynamics are unknown. We can achieve this by combining the proposed system with model-free reinforcement learning algorithms that allow online estimation of the value function [32]. In Fig. 3.5, we see the simulation results of the aforementioned procedure.

We note that the integral Bellman error becomes zero once the learning process has ended. Consequently, while it is possible to implement model-free secure optimal feedback policies, extensive research must be conducted on the capabilities of the attacker in compromising the learning process itself. This research direction will investigate a type of adversarial learning for CPS.

It is important for the network security community to derive a systematic way of evaluating the effect of proactive defense systems. So far, the usual procedure entails designing the MTD scheme and utilizing simulation results to show the effects on the attacker as well as to evaluate the overall optimality of the secure system. In this chapter, we present preliminary results that support the use of proactive security. Figure 3.6 shows the evolution of the data-driven intrusion detection system. In Fig. 3.7, we show the optimality loss that occurs when more weight is given to unpredictability during the entropy optimization. The theoretical bound is derived

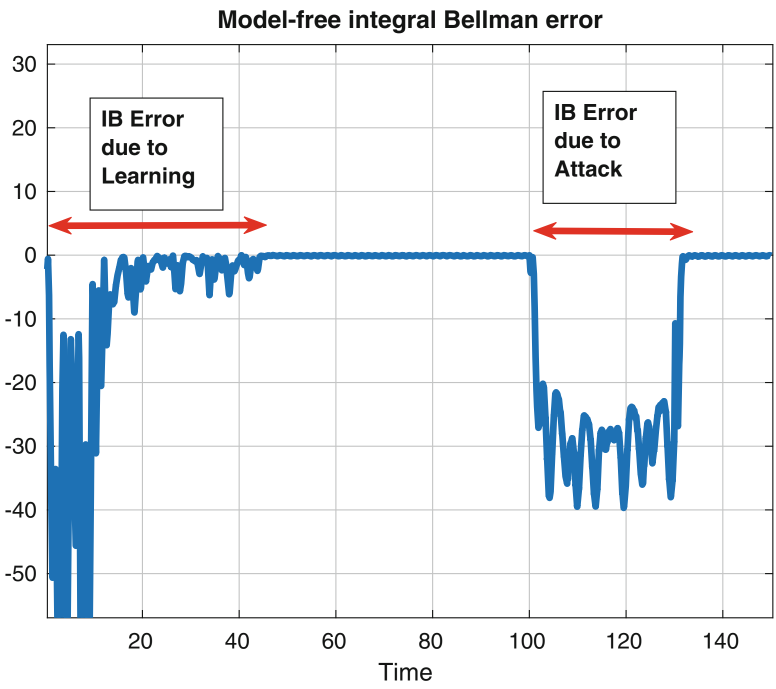


Fig. 3.6 The evolution of the model-free integral Bellman (IB) error combined with a reinforcement learning-based optimal policy

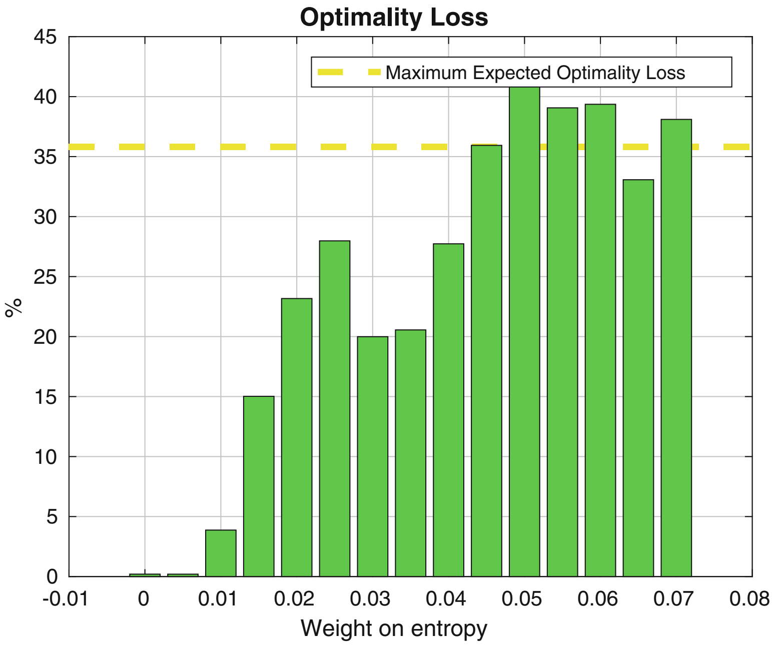


Fig. 3.7 Optimality loss induced by the unpredictable policies for different entropy levels

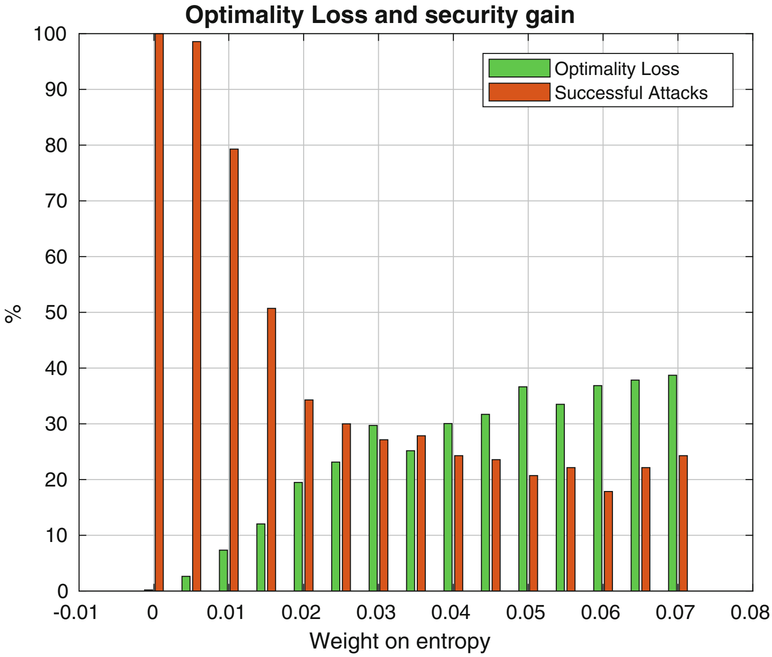


Fig. 3.8 Optimality loss and rate of successful attacks as a function of the entropy

when the active feedback policies follow a uniform distribution. In Fig. 3.8, we show the simultaneous increase in optimality loss, with the decrease in the percentage of successful defense penetrations for a given attack policy. It is noticeable that the rate with which the percentage of successful attacks decreases is greater than the optimality loss that the system suffers. This is a clear indication that proactive defense frameworks can indeed support the safe operation of the CPS.

However, those approaches rely on heuristic metrics. To examine the effects of different security algorithms rigorously, a realistic model of the behavior of the adversary is needed. In the literature, game-theoretic frameworks have been utilized in this regard. The infinite intelligence assumption that is central in game-theoretic results fails to take into account the bounded resources, both physical and cognitive, upon which the success of MTD rests. To mitigate that, novel models for malicious agents in networks must be introduced. The resulting non-equilibrium security game theory would allow the inclusion of deception techniques like MTD.

3.7 Conclusion and Future Work

In this chapter, a defense framework for CPS is developed. In order to increase the cost of the attack for the malicious agent, the system utilizes a proactive security mechanism based on the principles of MTD. A formal unpredictability metric is introduced and optimized so that the resulting switched system provides the best compromise between overall optimality and security. Simultaneously, the system's performance, quantified via the integral Bellman equation, is evaluated online and employed to detect intrusions in both the actuators and the sensors. Once an attack has been detected, the corresponding subsystems are isolated. Consequently, the CPS integrated with the proposed system can operate if the attacker has not compromised every available component.

Future work will include a rigorous analysis of the model-free case that was highlighted in Sect. 3.6. Also, we will examine the effect of attacks during the learning phase, formulating a continuous-time decision-theoretic equivalent of adversarial learning. Finally, in order to better evaluate the success of the proposed MTD algorithm, models of attacker intelligence will have to be introduced in our network security framework. Those models will be realistic in their reasoning and cognitive bounds.

Acknowledgements This work was supported in part by NATO under grant No. SPS G5176, by ONR Minerva under grant No. N00014-18-1-2160, and by an NSF CAREER under grant CPS-1851588.

References

1. R.R. Rajkumar, I. Lee, L. Sha, J. Stankovic, Cyber-physical systems: the next computing revolution, in *Proceedings of the 47th Design Automation Conference* (ACM, New York, 2010), pp. 731–736
2. J. Kim, H. Kim, K. Lakshmanan, R.R. Rajkumar, Parallel scheduling for cyber-physical systems: analysis and case study on a self-driving car, in *Proceedings of the ACM/IEEE 4th International Conference on Cyber-Physical Systems* (ACM, New York, 2013), pp. 31–40
3. I. Lee, O. Sokolsky, Medical cyber physical systems, in *Design Automation Conference (DAC), 2010 47th ACM/IEEE* (IEEE, Piscataway, 2010), pp. 743–748
4. Y. Mo, T.H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, B. Sinopoli, Cyber-physical security of a smart grid infrastructure. Proc. IEEE **100**(1), 195–209 (2012)
5. Y. Liu, P. Ning, M.K. Reiter, False data injection attacks against state estimation in electric power grids. ACM Trans. Inf. Syst. Secur. **14**(1), 13 (2011)
6. J. Slay, M. Miller, Lessons learned from the Maroochy water breach, in *International Conference on Critical Infrastructure Protection* (Springer, Berlin, 2007), pp. 73–82
7. J.P. Farwell, R. Rohozinski, Stuxnet and the future of cyber war. Survival **53**(1), 23–40 (2011)
8. S. Jajodia, A.K. Ghosh, V. Swarup, C. Wang, X.S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, vol. 54 (Springer, New York, 2011)
9. B. van Lier, Can cyber-physical systems reliably collaborate within a blockchain? Metaphilosophy **48**(5), 698–711 (2017)
10. A.A. Cardenas, S. Amin, S. Sastry, Secure control: towards survivable cyber-physical systems, in *28th International Conference on Distributed Computing Systems Workshops, 2008. ICDCS'08* (IEEE, Piscataway, 2008), pp. 495–500
11. T. Alpcan, T. Başar, *Network Security: A Decision and Game-Theoretic Approach* (Cambridge University Press, Cambridge, 2010)
12. M. Pajic, J. Weimer, N. Bezzo, O. Sokolsky, G.J. Pappas, I. Lee, Design and implementation of attack-resilient cyberphysical systems: with a focus on attack-resilient state estimators. IEEE Control. Syst. **37**(2), 66–81 (2017)

13. F. Pasqualetti, F. Dorfler, F. Bullo, Control-theoretic methods for cyberphysical security: geometric principles for optimal cross-layer resilient control systems. IEEE Control. Syst. **35**(1), 110–127 (2015)
14. B. Satchidanandan, P.R. Kumar, Dynamic watermarking: active defense of networked cyber-physical systems. Proc. IEEE **105**(2), 219–240 (2017)
15. K.G. Vamvoudakis, H. Modares, B. Kiumarsi, F.L. Lewis, Game theory-based control system algorithms with real-time reinforcement learning: how to solve multiplayer games online. IEEE Control. Syst. **37**(1), 33–52 (2017)
16. K.G. Vamvoudakis, J.P. Hespanha, Cooperative Q-learning for rejection of persistent adversarial inputs in networked linear quadratic systems. IEEE Trans. Autom. Control **63**, 1018–1031 (2017)
17. G. Theodorakopoulos, J.S. Baras, On trust models and trust evaluation metrics for ad hoc networks. IEEE J. Sel. Areas Commun. **24**(2), 318–328 (2006)
18. S. Jajodia, A.K. Ghosh, V. Subrahmanian, V. Swarup, C. Wang, X.S. Wang, *Moving Target Defense II: Application of Game Theory and Adversarial Modeling*, vol. 100 (Springer, New York, 2012)
19. V. Casola, A. De Benedictis, M. Albanese, A multi-layer moving target defense approach for protecting resource-constrained distributed devices, in *Integration of Reusable Systems* (Springer, Cham, 2014), pp. 299–324
20. J.H. Jafarian, E. Al-Shaer, Q. Duan, Openflow random host mutation: transparent moving target defense using software defined networking, in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks* (ACM, New York, 2012), pp. 127–132
21. M. Dunlop, S. Groat, W. Urbanski, R. Marchany, J. Tront, Mt6d: a moving target ipv6 defense, in *Military Communications Conference, 2011-Milcom 2011* (IEEE, Piscataway, 2011), pp. 1321–1326
22. R. Zhuang, S.A. DeLoach, X. Ou, Towards a theory of moving target defense, in *Proceedings of the First ACM Workshop on Moving Target Defense* (ACM, New York, 2014), pp. 31–40
23. Q. Zhu, T. Başar, Game-theoretic approach to feedback-driven multi-stage moving target defense, in *International Conference on Decision and Game Theory for Security* (Springer, Berlin, 2013), pp. 246–263
24. S. Weerakkody, B. Sinopoli, Detecting integrity attacks on control systems using a moving target approach, in *IEEE 54th Annual Conference on Decision and Control (CDC), 2015* (IEEE, Piscataway, 2015), pp. 5820–5826
25. H. Fawzi, P. Tabuada, S. Diggavi, Secure estimation and control for cyber-physical systems under adversarial attacks. IEEE Trans. Autom. Control **59**(6), 1454–1467 (2014)
26. Y. Yan, P. Antsaklis, V. Gupta, A resilient design for cyber physical systems under attack, in *American Control Conference (ACC), 2017* (IEEE, Piscataway, 2017), pp. 4418–4423
27. L. An, G.-H. Yang, Secure state estimation against sparse sensor attacks with adaptive switching mechanism. IEEE Trans. Autom. Control **63**, 2596–2603 (2017)
28. H. Okhravi, T. Hobson, D. Bigelow, W. Streilein, Finding focus in the blur of moving-target techniques. IEEE Secur. Priv. **12**(2), 16–26 (2014)
29. T.M. Cover, J.A. Thomas, *Elements of Information Theory* (Wiley, Hoboken, 2012)
30. J.P. Hespanha, A.S. Morse, Stability of switched systems with average dwell-time, in *Proceedings of the 38th IEEE Conference on Decision and Control, 1999*, vol. 3 (IEEE, Piscataway, 1999), pp. 2655–2660
31. A. Kanellopoulos, K.G. Vamvoudakis, Switching for unpredictability: a proactive defense control approach, to appear in, American Control Conference, Philadelphia, PA (2019)
32. D. Vrabie, K.G. Vamvoudakis, F.L. Lewis, *Optimal Adaptive Control and Differential Games by Reinforcement Learning Principles*, vol. 2. IET (2013)

33. R.J. Martin, L. Valavani, M. Athans, Multivariable control of a submersible using the lqg/ltr design methodology, in *American Control Conference, 1986* (IEEE, Piscataway, 1986), pp. 1313–1324
34. X. Yu, J. Jiang, Hybrid fault-tolerant flight control system design against partial actuator failures. IEEE Trans. Control Syst. Technol. **20**(4), 871–886 (2012)

Chapter 4
Security-Aware Incentives Design for Mobile Device-to-Device Offloading

Jie Xu and Lixing Chen

Abstract Device-to-Device (D2D) computation offloading, or D2D offloading, exploits spare computing resources of nearby user devices to enhance mobile computing performance. Its success relies on user participation in costly collaborative service provisioning, thus mandating an incentive mechanism that can compensate for these costs. Although incentive mechanism design has been studied extensively in the literature, this paper considers a more challenging yet less investigated problem in which selfish users are also facing interdependent security risks that depend on the collective behavior of all users. To this end, we build a novel mathematical framework by combining the power of game theory and epidemic theory to investigate the interplay between user incentives and interdependent security risks in D2D offloading, thereby enabling the design of security-aware incentive mechanisms. Our analysis discovers an interesting "less is more" phenomenon: although giving users more incentives promotes more participation, it may harm the network operator's utility. This is because too much participation may foster persistent security risks and as a result, the effective participation level does not improve.

4.1 Introduction

More and more mobile applications nowadays demand resources (e.g. processer power, storage, and energy) that frequently exceed what a single mobile device can deliver. To meet this challenge, mobile cloud computing (MCC) ([1] and references therein) offloads computation tasks from mobile devices to the remote cloud for processing, and more recently, mobile edge computing (MEC) [2, 3] moves computation and storage capabilities from the central cloud to the network edge devices such as base stations (BSs). In both MCC and MEC, mobile users access

J. Xu (✉) · L. Chen
University of Miami, Coral Gables, FL, USA
e-mail: jiexu@miami.edu; lx.chen@miami.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_4

computing services on a centralized and fixed server through wireless connection. However, the ever-growing number of mobile devices and volume of sensory data pose increasingly heavy traffic burden on the core wireless network. Moreover, edge servers are also likely to be overloaded due to the increasing computation demand. To overcome these drawbacks, device-to-device (D2D) communication is exploited for offloading computation to nearby peer mobile devices, thereby fully unleashing the potential of their computation power [4–7]. This technique, known as D2D offloading, not only increases the wireless network capacity but also alleviates the computation burden from centralized servers.

However, the mobile D2D architecture poses many challenges on computation offloading and two key challenges are incentives and security: because providing offloading service incurs extra costs (e.g. computation/transmission energy consumption) to the mobile users acting as servers, incentives must be devised to encourage selfish users to participate; because D2D offloading relies on ordinary mobile users whose security protection is much weaker than the centralized server, D2D offloading is much more vulnerable to security attacks [8]. A commonly seen security risk in mobile networks is proximity-based infectious attacks [9, 10]. In such attacks, mobile user equipments can get compromised by proximity-based mobile viruses when they are acting as D2D servers and consequently are unable to provide service to other users. Moreover, compromised users become new sources of attacks when they communicate to other users via D2D in the future, thereby spreading the attack across the entire network. Although incentives and security are often treated as separate topics in the literature, they are indeed intricately intertwined in the mobile D2D offloading system. On the one hand, the outcome of attack depends on users' collective decisions on the participation in D2D offloading and hence, the risk is interdependent among all users. Intuitively, more participation fosters faster and wider spread of the attack and hence may cause a greater damage to the overall system. On the other hand, the security risk shapes individual users' participation incentives. In view of a larger chance of being compromised, individual users may strategically reduce their participation levels, thereby degrading the performance of collaborative computation. Although there is a huge literature on incentive mechanism design, systematic understanding of participation incentives under interdependent security risks and their impacts on the network performance is largely missing.

In this chapter, we investigate the interplay between incentives and interdependent security risks in mobile D2D offloading and design security-aware incentive mechanisms. To this end, we build a novel analytical framework by combining the power of game theory [11] and epidemic theory [12]. Since our main focus is the potential infection risks due to D2D interactions (i.e. traffic exchange), the developed techniques can also be extended to other mobile D2D systems besides computation offloading. The main contributions of this work are as follows:

1. We focus on the utility maximization of a wireless network operator who provides both communication and computation services. The problem is formulated as a Stackelberg game in which the operator is the leader, who designs the

incentive mechanism for D2D offloading, and the mobile users are the followers, who decide their participation levels by playing a non-cooperative game among themselves in the presence of interdependent security risks.

2. We characterize individual users' participation incentives under infectious proximity-based risks with tunable user foresightedness. Users' participation strategy is shown to have a threshold structure: a user is willing to provide wireless D2D offloading service if and only if the risk (i.e. the chance of serving a compromised requester) is low enough.
3. We analyze the infection propagation dynamics when selfish users are strategically determining their participation levels. This analysis is in stark contrast with existing epidemic studies which assume non-strategic users obediently following prescribed and fixed actions. A key result is a phase transition between persistent and non-persistent infection, which is substantially different in nature from the non-strategic user case.
4. We discover an interesting "Less is More" phenomenon: although offering users a higher reward promotes participation, a too high participation level degrades the system performance and consequently reduces the operator's utility. This is because too much participation fosters *persistent* infection and hence, the *effective* participation level does not improve.

4.2 Related Work

Data and computation offloading is used to address the inherent problems in mobile computing by using resource providers other than the mobile device itself to host the execution of mobile applications [13]. Our work focuses on D2D offloading which uses nearby mobile devices as resource providers [4, 5, 14]. In general, computation offloading is concerned with what/when/how to offload a user's workload from its device to other resource providers (see [15, 16] and references therein). The problem studied in our work is not contingent on any specific offloading technology. Rather, we design incentives in the presence of interdependent security risks and hence, our approach can be used in conjunction with any existing offloading technology.

D2D offloading benefits from the fact that mobile users in close proximity can establish direct wireless communication link over the licensed spectrum (*inband*) or unlicensed spectrum (*outband*) while bypassing the cellular infrastructure such as the BSs [17–19]. In practice, D2D communication has been implemented in industry products such as Qualcomm FlashLinQ [20] and standardized by 3GPP [21]. Enabled by D2D communication, D2D offloading can further alleviate computation burdens from the edge computing infrastructure [22]. A general consensus in the literature is that mobile users would have no incentive to participate in D2D service provisioning unless they receive satisfying rewards from the network operator [23]. Our prior works [24, 25] design virtual currency-based incentive mechanism to promote wireless D2D relaying. A contract-based incentive mechanism is developed in [26] to let users self-reveal their preferences. Market models are developed in [23] in which Stackelberg games and auctions are used to design incentive mechanisms for open markets and sealed markets, respectively. However, these existing works do not consider the interdependent security risks in the D2D interactions.

One of the greatest challenges for D2D offloading is the interdependent security risk among users. Although offloading in general faces many security and privacy issues, this work focuses on proximity-based infectious attacks which rely on the cooperative interaction among users. Extensive research has shown that such attacks can be easily initiated in wireless mobile networks and cause significant damage [9, 10]. To model such attacks, we utilize the popular Susceptible-Infected-Susceptible (SIS) model [27–30] from the epidemics research community, which is a standard stochastic model for virus infections and widely-adopted to investigate computer virus/worm propagation. Existing works in this regard can be divided into two categories. The first category adopts a mean-field approximation to study networks consisting of a large number of individuals [27, 29]. The second category tries to understand the influence of graph characteristics on epidemic spreading when users are interacting over a fixed topology [28, 30]. Since mobile users are numerous and server-requester matching is short-lived and random due to user mobility and task arrival processes, the mean-field approach provides a good model approximation for the D2D offloading architecture. However, all these works study *non-strategic* users who are following given and fixed actions. The present work significantly departs from this strand of literature and studies *strategic* users who choose their participation levels to maximize their own utility.

Stackelberg game formulations have been proposed in resource allocation problems in wireless networks and D2D communication systems [31, 32]. For instance, in [31], the D2D pairs as followers compete selfishly for the available bandwidth in a non-cooperative game after the BS as the leader establishes a set of "prices" for the received interference power from the D2D transmission on each subchannel. Our considered problem is not a resource allocation problem. We study how to incentivize users to participate in the D2D system and the formulated Stackelberg game is used to understand users' incentives under infectious risks and determine the optimal incentive strategy by the operator.

4.3 System Model

4.3.1 Network Model and Incentive Mechanism

We consider a continuous time system and a wireless network in which mobile user equipments (UEs) generate computation tasks over time. We adopt a continuum population model for UEs to capture the large number of UEs in the network. The network operator provides computation service so that UEs can offload their data and tasks to the edge/cloud servers. When the edge/cloud server reaches its computation capacity or the wireless network is congested, the operator will try to employ D2D offloading, if possible, as a supplement in order to fulfill UEs'

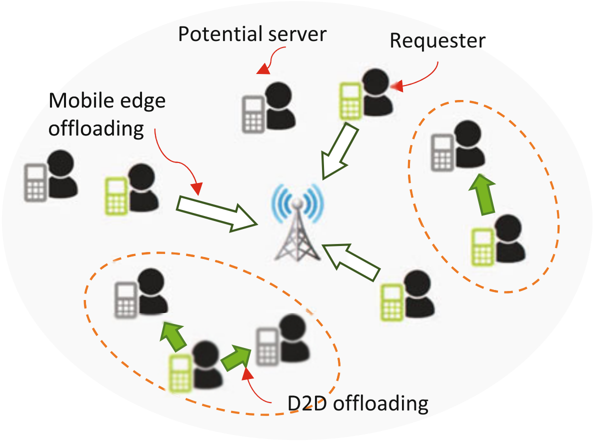


Fig. 4.1 Snapshot of one of the cells in the network

computation requests. In this case, the task data will be transmitted to nearby UEs that have spare computing resources via wireless D2D communication (e.g. Wifi-Direct [33] or LTE-Direct [34]). To facilitate the exposition, we call the UE who requests the offloading service as the *requester* and the UE who provides the service as the *server*. The considered wireless network is *dynamic* in two senses. First, UEs are moving in the network and hence the physical topology of the network is changing. We consider a generic mobility model that results in strongly mixed user population. Second, each UE can be a requester when it has demand and can also be a server when it is idle. As a result, the matchings of requesters and servers are also changing over time depending on both the physical topology and the demand arrival process. Note that there could be a lot of concurrent D2D offloading instances going on at the same time in the network. Figure 4.1 shows a snapshot of part of the network.

For each task completed via D2D offloading, the operator obtains a benefit due to the saved wireless bandwidth and computation cost. The expected value of this benefit is denoted by b_0. On the other hand, D2D offloading incurs an extra cost to the UE acting as the D2D server due to computation and transmission energy consumption and hence, selfish UEs are reluctant to provide the D2D service unless proper incentives are provided. Let the expected cost incurred to UE i by completing one task be c_i, which differs across UEs. Although the realized cost depends on the specific computation task and the instantaneous wireless channel condition, for the UE's decision making purpose, we consider only the expected cost. In order to motivate participation in providing D2D offloading service, the operator offers reward to the participating UEs, in forms of free data or monetary payments, and how much reward a UE can receive depends on its participation level.

We use contracts as the incentive mechanism as in [26]. Specifically, each UE i makes a participation-reward bundle contract $(a_i^t, r(a_i^t))$ with the network operator at any time t where a_i^t is the participation level chosen by UE i, and $r(a_i^t)$ is the unit time reward offered by the operator. A contract $(a_i^t, r(a_i^t))$ requires that UE i provides D2D offloading service with a rate up to a_i^t tasks (from possibly different requesters) per unit time. The reward $r(a)$ is an increasing function of a. As a practical scheme, the operator adopts a throttled linear reward function

$$r(a) = \begin{cases} r_0 a, & \forall a \leq R_{max}/r_0 \\ R_{max}, & \forall a > R_{max}/r_0 \end{cases} \tag{4.1}$$

where r_0 is the unit reward and R_{max} is the maximum reward that the operator is willing to pay. Let $M \triangleq R_{max}/r_0$. Such a throttled scheme enables easy system implementation and similar schemes are widely adopted by operators in the real world.[1] Nevertheless, our framework and analysis can also be applied to the reward scheme without throttling. In practice, contracts have a minimum duration. To simplify our analysis, we assume that the minimum contract duration is small enough so that UEs can effectively modify their contracts at any time.

4.3.2 Participation Incentives in the Attack-Free Network

The operator assigns computation tasks to UEs according to their chosen participation levels whenever D2D offloading is needed. Due to UE mobility and the randomness in the task arrival, we model the D2D offloading request arrival assigned to UE i as a Poisson process with rate a_i^t (which is a result of UE decision). The unit time utility function of UE i by choosing a participation level a_i^t is thus defined as

$$u_i(a_i^t) = v_i(r_0 a_i^t) - c_i a_i^t \tag{4.2}$$

where $v_i(\cdot)$ is UE i's evaluation function of the reward, which differs across UEs. Clearly, selfish UEs have no incentives to participate at a level greater than M if the reward is throttled and hence, we will focus on the range $a_i^t \in [0, M]$. We make the following assumptions on $v(\cdot)$.

Assumption 1 (1) $v_i' > 0$, $v'' < 0$. (2) $v(0) = 0$, $v_i(r_0 M) > c_i M$. (3) $v_i'(r_0 M) < c_i/r_0 < v_i'(0)$.

Part (1) is the standard diminishing return assumption. It means that more reward has a higher value to the UE; however, the marginal value decreases as the reward increases. Part (2) states that zero participation brings zero utility, namely $u_i(0) = 0$, and the maximum participation yields at least a positive utility, namely

[1] For example, ATT has a data reward program [35] in which users earn extra data by downloading games and apps or shopping in participatory stores with a data transfer capping of 1000 MB per bill period.

$u_i(M) > 0$. Part (3) assumes that the optimal participation level lies in $(0, M)$. Because $v_i'(0) > c_i/r_0$, $u_i'(0) = r_0 v_i'(0) - c_i > 0$. Because $v_i'(r_0 M) < c_i/r_0$, then $u_i'(M) = r_0 v_i'(r_0 M) - c_i < 0$. This means we must have a solution $a_i^{\mathrm{AF}}(r_0) \in (0, M)$ so that $u_i'(a_i^{\mathrm{AF}}) = 0$, which is the optimal participation level in the attack free (AF) network.

We make a further assumption on $a_i^{\mathrm{AF}}(r_0)$ as follows.

Assumption 2 $a_i^{\mathrm{AF}}(r_0)$ is increasing in r_0.

Assumption 2 states that increasing reward provides UEs with more incentives to participate as a D2D server. This is a natural assumption and can be easily satisfied by many evaluation functions. For instance, if $v_i(x) = \sqrt{x}$, then $a_i^{\mathrm{AF}}(r_0) = \frac{r_0}{4c_i^2}$ which is increasing in r_0.

4.3.3 Attack Model

Participating as D2D servers exposes UEs to proximity-based infectious risks since the task data is sent directly from peer UEs via D2D communication rather than the trusted operator-owned BSs. We consider an attack whose purpose is to make the D2D offloading service unusable. Therefore compromised UEs will not be able to provide D2D offloading service. Moreover, they may become new sources of attack when they request offloading services from normal UEs in the future. To model this attack, we adopt the popular Susceptible-Infected-Susceptible (SIS) epidemic model. At any time t, each UE i is in one of the two states $s_i^t \in$ {(S)usceptible, (I)nfected}, which indicates whether the UE is normal or compromised. The UE state is private information and unknown to either the operator or other UEs in the network. Otherwise, compromised UEs can be easily excluded. If a normal UE provides D2D offloading service to a compromised UE, then the normal UE gets infected by the virus with a probability $\beta \in [0, 1]$. We assume that a compromised server does not infect (or with a negligible probability infects) a normal requester because detection is much more effective on the requester side due to the significantly smaller data size of the returned computation result [2].

Once a UE i is compromised, it has to go through a recovery process. During this process, UE i cannot provide any D2D offloading service to other UEs. Moreover, UE i suffers a recovery cost q_i per unit time. However, UE i can still make offloading requests so that the virus can be propagated to other UEs. Nevertheless, our framework can be easily extended to handle the case where some compromised UEs are completely down so that they cannot make requests before recovery. We assume that the recovery process duration is exponentially distributed with mean $1/\delta$. Recovered UEs re-enter the normal state and may be compromised again in the future. The UE state transition is illustrated in Fig. 4.2.

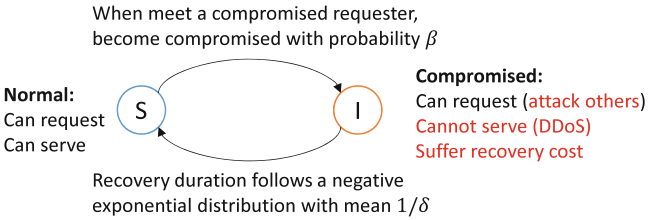

When meet a compromised requester, become compromised with probability β
Normal:
Can request
Can serve
S
I
Compromised:
Can request (attack others)
Cannot serve (DDoS)
Suffer recovery cost
Recovery duration follows a negative exponential distribution with mean $1/\delta$

Fig. 4.2 UE state transition

The parameters β and δ describe the intrinsic risk level of the wireless network and we define the *effective infection rate* as $\tau \triangleq \beta/\delta$. We will treat these risk parameters as fixed for most part of this chapter. In this way, we focus on the incentive mechanism design for a network in a given risk environment. Later, we will discuss how the incentive mechanism and security technologies can be jointly designed.

4.3.4 Problem Formulation as a Stackelberg Game

The objective of the operator is to design the unit reward r_0 in order to maximize its own utility. The operator's utility is defined as:

$$u_0(r_0) = \mathbb{E}_{t,i}[(b_0 - r_0)\mathbf{1}\{s_i^t = S\}a_i^t] \tag{4.3}$$

where the expectation is taken over the UE attributes (i.e. the distribution of the evaluation function $v_i(\cdot)$ and D2D offloading cost c_i) and time. Clearly we must have $r_0 < b_0$. Otherwise, the operator will not adopt D2D offloading. The incentive mechanism design problem can be formulated as a Stackelberg game among a leader and an infinite number of followers (due to the continuum population model). The operator plays as the leader, which moves first and determines the reward mechanism r_0. The UEs play as the followers, which move next and choose their participation levels. In the attack-free network, the Stackelberg game can be represented by the following two-level optimization problem

$$\max_{r_0} \ (b_0 - r_0)\mathbb{E}_i[a_i^{\text{AF}}(r_0)] \ \textit{subject to} \ a_i^{\text{AF}}(r_0) = \arg\max_a u_i(a|r_0), \forall i \tag{4.4}$$

where $\mathbb{E}_{t,i}[\mathbf{1}\{s_i^t = S\}a_i^t]$ is replaced with $\mathbb{E}_i[a_i^{\text{AF}}(r_0)]$ since there is no attack and hence the UEs are never compromised. This problem is not difficult to solve as $a_i^{\text{AF}}(r_0)$ can be easily computed.

The presence of infectious attacks, however, changes both the operator's objective function and UEs' participation incentives. First, because UEs may get compromised and consequently are not able to provide D2D computing service sometimes, the utility that the operator can reap from D2D offloading depends on not only the UEs' participation decisions but also the network security state (i.e. the fraction of normal UEs). Therefore, the operator's utility becomes $(b_0 - r_0)\mathbb{E}_i[\mathbf{1}\{s_i^t = S\}a_i^*(r_0)]$ and we call $\mathbb{E}_i[\mathbf{1}\{s_i^t = S\}a_i^*(r_0)]$ the *effective* participation level of D2D offloading. Second, UE i's incentive to participate in D2D offloading will be determined by a utility function $U_i(a_i)$ that incorporates the potential future infection (which will be characterized in the next section). What much complicates the problem is the fact that UEs face interdependent security risks—the security posture of the whole network depends on not only the participation action of UE i itself but also all other UEs' participation since the infection is propagated network-wide. Therefore, the utility function $U_i(a_i)$ should indeed be a function of all UEs' actions and hence, we denote it by $U_i(a_i, \boldsymbol{a}_{-i})$ where $\boldsymbol{a}_{-i}$, by convention, is the action profile of UEs except i. Formally, the Stackelberg game for the security-aware incentive mechanism design problem is:

$$\max_{r_0} \ (b_0 - r_0)\mathbb{E}_i[\mathbf{1}\{s_i^t = S\}a_i^*(r_0)] \ \textit{subject to} \ \ a_i^*(r_0) = \arg\max_{a} U_i(a_i, \boldsymbol{a}_{-i}|r_0), \forall i \tag{4.5}$$

In the above Stackelberg game, the followers (UEs) not only perform best response to the leader (i.e. the operator)'s decision, but also play a different yet coupled game among themselves due to the interdependent security risk. To solve this game, we will use backward induction to firstly investigate the participation incentives of UEs under the interdependent security risk and the resulting *effective* participation level $\mathbb{E}_i[\mathbf{1}\{s_i^t = S\}a_i^*(r_0)]$, and then design the optimal reward mechanism.

4.4 Individual Participation Incentives

4.4.1 Foresighted Utility

If a UE is myopic and only cares about the instantaneous utility, then the UE will simply choose a participation level $a_i^* = \arg\max_a [v_i(r_0 a) - c_i a]$ which maximizes the instantaneous utility when it is in the normal state. In such cases, $a_i^* = a^{AF}$. However, since the infectious attack may cause potential future utility degradation, the UE will instead be foresighted and care about the *foresighted utility* [36]. Since the infection risk depends on the probability that a server UE meets a compromised requester UE, the fraction of compromised UE in the system at any time t, denoted by $\theta^t \in [0, 1]$, plays a crucial role in computing the foresighted utility. Assume that requester-server matching for D2D offloading is uniformly random, then the probability that server UE i meets a compromised requester UE is exactly θ^t. We thus call θ^t the *system compromise state* at time t. Note that θ^t is an outcome of all UE's participation decisions. The foresighted utility is defined as follows.

Definition 4.1 (Foresighted Utility) Given the system compromise state θ, the foresighted utility of UE i with discount rate ρ when it takes a participation action a_i is defined as

$$U(a_i, \theta) = \rho \int_{t=0}^{\infty} (\underbrace{\int_{\tau=0}^{t} e^{-\rho\tau} u_i(a_i) d\tau}_{\text{discounted sum utility before being infected}} + \underbrace{\rho^{-1} e^{-\rho t} U_I}_{\text{discounted continuation utility after being infected}}) \underbrace{\theta\beta a_i e^{-\theta\beta a_i t}}_{\text{exponential distribution due to Poisson arrival}} dt \tag{4.6}$$

where

$$U_I = \rho \int_{t=0}^{\infty} \left(\int_{\tau=0}^{t} e^{-\rho\tau}(-q_i) d\tau + \rho^{-1} e^{-\rho t} U(a_i, \theta) \right) \delta e^{-\delta t} dt \tag{4.7}$$

We explain the definition of foresighted utility below:

(i) Consider a UE who decides its current contract at time t_0. It stays in the normal state until the next time (say $t_0 + t$) when (1) it interacts with a compromised UE, (2) the attack by the compromised UE is successful and (3) the UE has not been compromised during $[t_0, t_0 + t)$. This is a random process (where t is the random variable). The probability density function of being compromised at time $t_0 + t$ is $\theta\beta a_i e^{-\theta\beta a_i t}$.

(ii) Suppose that UE i is compromised at time $t_0 + t$, then $\int_{\tau=0}^{t} e^{-\rho\tau} u_i(a_i) d\tau$ is the discounted sum utility that the UE can receive during the period $[t_0, t_0 + t]$ before it is compromised. The term $e^{-\rho\tau} \le 1$ represents the discounting mechanism which decreases with τ. A larger discount rate ρ means that the discounting is greater. The utility discounting mechanism is widely used in the literature [36, 37] when modeling user foresightedness to capture the fact that users often value the current utility more than the future utility, and the exponential function $e^{-\rho\tau}$ is the standard method for continuous time systems. It not only offers mathematical tractability (because simply summing up utility without discounting over an infinite horizon can result in an unbounded utility value) but also is well-motivated by real-world considerations. For example, when the utility involves monetary payments, the discount factor is directly related to the interest rate. Moreover, discounting can also occur if users may leave the system and receive 0 utility after leaving the system. Assume that a UE stays in the system following an exponential distribution with mean $1/\rho$, then the probability that it is still in system by time $t_0 + \tau$ conditional on that the UE is in the system at time t_0 is $e^{-\rho\tau}$. Therefore the utility is discounted by $e^{-\rho\tau}$.

(iii) U_I is the utility that UE i receives once it gets compromised. During the recovery phase, UE i suffers a cost $-q_i$ and once it is recovered, it receives again the foresighted utility $U(a_i, \theta)$ which is discounted by $e^{-\rho t}$ where t the duration of the recovery phase. Here we assumed *bounded rationality*: when computing the foresighted utility, the UE believes that it will choose the *same* participation level as before once it is recovered since it expects the system

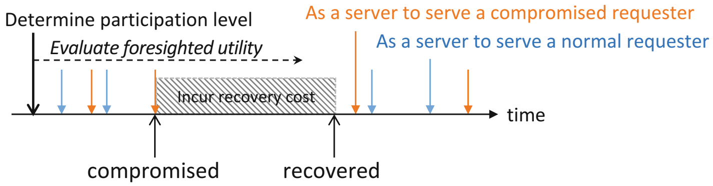

Determine participation level
Evaluate foresighted utility
As a server to serve a compromised requester
As a server to serve a normal requester
Incur recovery cost
time
compromised
recovered

Fig. 4.3 Illustration of foresightedness in participation level determination

compromise state to stay the same in the future. In the steady state, this belief will in fact be correct.

(iv) The constant ρ at the beginning of the right-hand side equations is a normalization term, which is due to $\int_{t=0}^{\infty} e^{-\rho t} dt = \rho^{-1}$.

Figure 4.3 illustrates the role of foresightedness plays in determining the participation level of a UE. By solving (4.6) and (4.7), the foresighted utility can be simplified to

$$U(a_i, \theta) = \frac{(\rho + \delta)u_i(a_i) - \beta\theta a_i q}{\rho + \delta + \beta\theta a_i} \tag{4.8}$$

The above form of foresighted utility generalizes the instantaneous utility and the time-average long-term utility, and can capture a wide spectrum of UE behaviors by tuning the discount rate ρ. When $\rho \to \infty$, the UE cares about only the instantaneous utility. In this case, $U(a_i, \theta)$ reduces to the myopic utility $u(a_i)$. When $\rho \to 0$, the UE does not discount at all and hence, $U(a_i, \theta)$ becomes the time-average utility $\frac{\delta u(a_i) - \beta\theta a_i q}{\delta + \beta\theta a_i}$, which is the same result by performing the stationary analysis of a continuous-time two-state Markov chain.

4.4.2 Individual Optimal Participation Level

In this subsection, we study the optimal participation level that UE i will choose to maximize its foresighted utility.

Proposition 4.1 *If the system compromise state* $\theta \geq \frac{(r_0 v_i'(0) - c_i)(\rho + \delta)}{q_i \beta} \triangleq \bar{\theta}_i$, *then the optimal participation level is* $a_i^*(\theta) = 0$. *Otherwise, the optimal participation levela*$_i^*(\theta)$*is the unique solution of*$u_i'(a_i)(\rho + \delta\beta\theta a_i) - u_i(a_i)\beta\theta - \beta\theta q_i = 0$, *which increases as* θ *decreases.*

Proof See Appendix.

Proposition 4.1 can be intuitively understood. A larger system compromise state θ implies a higher risk of getting compromised via D2D offloading and hence, UE i has smaller participation incentives. In particular, if the system compromise state exceeds a threshold, then UE i will refrain from participating in D2D offloading. It is also evident that the operator can provide UEs with more incentives to participate by increasing the reward r_0.

Proposition 4.2 *If $\theta < \bar{\theta}_i$, then $a_i^*(\theta)$ is increasing in ρ, δ and decreasing in β.*

Proof These claims are direct results of the monotonicity of the left-hand side of (4.20).

Proposition 4.2 states that a higher attack success probability (larger β) and a slower recovery speed (smaller δ) both decrease UE i's incentives to participate (smaller α^*). Importantly, Proposition 4.2 also reveals the impact of UE foresightedness on the participation incentives: being more foresighted (smaller ρ) decreases the UE's participation incentives (smaller α^*) because the UE cares more about the potential utility degradation caused by the attack.

4.5 Interdependent Security Risks

In this section, we study how the infection propagates in the system and the convergence of the system compromise state. Epidemic processes have been well investigated in the literature for different systems. Most of this literature assumes that users are obediently following a prescribed behavior rule. However, selfish UEs in the considered problem determine their participation levels to maximize their own foresighted utilities, thereby leading to significantly different results than conventional epidemic conclusions. To show this difference, we will first review the classic results of infection propagation in the context of the considered D2D offloading network. Then we will study the infection propagation processes for selfish UEs in two scenarios. In the first scenario, UEs observe the system compromise state at any time and hence make adaptive decisions according to the observed state. In the second scenario, UEs do not observe the system compromise state and hence have to conjecture this state based on the equilibrium analysis of a D2D offloading participation game.

4.5.1 Attack Under Given Participation Actions

The evolution of the system compromise state θ^t depends on the strategy adopted by the UEs, the attack success rate β, the recovery rate δ as well as the initial state of the system θ^0. It is obvious that if the system starts with an initial state $\theta^0 = 0$ (i.e.

without initial attacks), then the system will remain in the state of zero compromise no matter what strategies are adopted by the UEs. Therefore, we will focus on the non-trivial case that the initial state $\theta^0 > 0$.

The network is said to be in the steady state if the system compromise state θ^t becomes time-invariant, denoted by θ^∞. The steady state reflects how the infectious attack evolves in the long-run. Existing works suggest that there exists a critical threshold τ_c for the effective infection rate τ such that if $\tau < \tau_c$, then the infectious attack extinguishes on its own even without external interventions, namely $\theta^\infty = 0$; otherwise, there is a positive fraction of compromised UEs, namely $\theta^\infty > 0$. This result is formally presented in our problem as follows.

Proposition 4.3 *Assume that all UEs adopt the same fixed participation level a. Then there exists a critical effective infection rate* $\tau_c = \frac{1}{a}$, *such that if* $\tau \le \tau_c$, $\theta^\infty = 0$; *otherwise*, $\theta^\infty = 1 - \frac{\delta}{\beta a}$.

Proof See Appendix.

Notice that if $\beta < \delta$, then for all values of a, we must have $\tau \le \tau_c$ and hence, the infection risk always extinguishes in the long-run.

4.5.2 Attack with Strategic UEs and Observable System Compromise State

In the considered D2D offloading system, UEs strategically determine their participation levels and hence, the infection propagation dynamics may be significantly different. In this subsection, we assume that UEs can observe the system compromise state θ^t at any time. In this case, UEs adaptively change their participation levels by revising their contracts with the operator according to the observed system compromise state.

For the analysis simplicity, we assume that all UEs have the same evaluation function $v(\cdot)$, service provision cost c and recovery cost q. We will investigate the heterogeneous case in the next subsection and in simulations. The system dynamics thus can be characterized by the following equation,

$$d\theta^t = -\theta^t \delta dt + (1 - \theta^t)\beta\theta^t a^*(\theta^t)dt \tag{4.9}$$

where $a^*(\theta^t)$ is the best-response participation level given the current system compromise state θ^t according to our analysis in the previous section. In the above system dynamics equation, the first term $\theta^t \delta dt$ is the population mass of compromised UEs that are recovered in a small time interval dt. The second term $(1 - \theta^t)\beta\theta^t a^*(\theta^t)dt$ is the population mass of normal UEs that are compromised in a small time interval dt. The following proposition characterizes the convergence of the system dynamics.

Proposition 4.4 *There exists a critical effective infection rate* $\tau_c = \frac{1}{a^{\mathrm{AF}}}$ *such that if* $\tau < \tau_c$*, then the system compromise state converges to* $\theta^{\infty} = 0$*; otherwise,* $\theta^{\infty} = \theta^{\dagger}$*where* $\theta^{\dagger} > 0$ *is the unique solution of* $(1 - \theta^{\dagger})a^{*}(\theta^{\dagger}) = 1/\tau$.

Proof See Appendix.

Proposition 4.4 shows that when UEs are selfish and strategically deciding their participation levels, the infectious attack propagation also has a thresholding effect with regard to the effective infection rate. However, this thresholding effect is significantly different in nature from when UEs are obeying prescribed participation actions. In the non-strategic case, the effective infection rate threshold is a function of the given participation level. In the strategic case, the threshold is a constant. In particular, the constant is exactly the critical threshold when UEs follow the individually optimal action a^{AF} in the attack-free network (see Fig. 4.4 for an illustration).

Let us understand this thresholding effect better. As proved in Proposition 4.1, the individual optimal participation level under infectious attack risks is always lower than that in the attack-free network. Therefore, although the participation level is adapting over time depending on the system compromise state, it will never be higher than a^{AF}. As a result of Proposition 4.3, if the effective infection rate is less than $1/a^{\mathrm{AF}}$, the attack will extinguish on its own. When the effective infection rate becomes sufficiently large (i.e. $\tau > 1/a^{\mathrm{AF}}$), individual UEs always have incentives to choose a participation level greater than the threshold participation level that eliminates infection. This is because unilaterally increasing the participation level does not change the network-wide epidemic dynamics since each individual UE is infinitesimal in the continuum population model yet increases the benefit that the individual UE can obtain. Therefore infectious attacks persist.

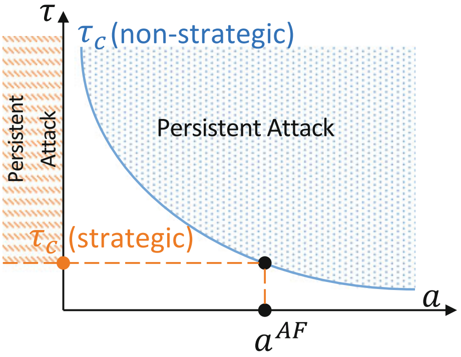


Fig. 4.4 Critical effective infection rates in strategic and non-strategic cases

4.5.3 Attack with Strategic UEs and Unobservable System Compromise State

In practice, neither the operator nor UEs observe the system compromise state in real time. In this case, UEs have to conjecture how the other UEs will participate in the D2D offloading system and the resulting system compromise state. To handle this situation, we formulate a population game to understand the D2D participation incentives under the infectious attacks. To enable tractable analysis, we assume that there are K types of UEs. This is not a strong assumption since we do not impose any restriction on the value of K. UEs of the same type k have the same evaluation function $v_{(k)}(\cdot)$, service provision cost $c_{(k)}$ and recovery cost $d_{(k)}$. Denote the fraction of type k UEs by w_k and we must have $\sum_{k=1}^{K} w_k = 1$.

In the D2D offloading participation game, each UE is a player who decides its participation level. Since UEs do not observe the system compromise state, it is natural to assume the each UE adopts a constant strategy, namely $a_i^t = a_i, \forall t$. The Nash equilibrium of the D2D participation game is defined as follows.

Definition 4.2 (Nash Equilibrium) A participation action profile $\boldsymbol{a}^{\text{NE}}$ is a Nash equilibrium if for all i, $a_i^{\text{NE}} = \arg\max_{a_i} U(a_i, \theta^{\infty}(\boldsymbol{a}^{\text{NE}}))$ where $\theta^{\infty}(\boldsymbol{a}^{\text{NE}})$ is the converged system compromise state under $\boldsymbol{a}^{\text{NE}}$.

First, we characterize the converged system compromise state assuming that all type k UEs choose a fixed participation level $a_{(k)}$.

Proposition 4.5 *Given the type-wise participation level vector* $(a_{(1)}, ..., a_{(K)})$*, there exists a critical effective infection rate* $\tau_c = \frac{1}{\sum_{k=1}^{K} w_k a_{(k)}}$*, such that if* $\tau \leq \tau_c$*,* $\theta^{\infty} = 0$*; otherwise,* $\theta^{\infty} > 0$ *is the unique solution of* $\sum_{k=1}^{K} \frac{\tau w_k a_{(k)}}{\tau \theta^{\infty} a_{(k)} + 1} = 1$.

Proof See Appendix.

Proposition 4.5 is actually the extended version of Proposition 4.3, which further considers heterogeneous UEs. In the homogeneous UE case, the critical infection rate depends on the homogeneous participation level of UEs. In the heterogeneous UE case, the critical infection rate depends on the average participation level of UEs. With Proposition 4.5 in hand, we are able to characterize the Nash equilibrium of the D2D participation game.

Theorem 4.1 *(1) The D2D participation game has a unique NE. (2) The NE is symmetric within each type, namely* $a_i^{\text{NE}} = a_{(k)}^{\text{NE}}$ *for all UE i with type k. (3) If* $\tau \leq \frac{1}{\sum_{k=1}^{K} w_k a_{(k)}^{\text{AF}}}$*, then* $\theta^{\infty} = 0$ *and* $a_{(k)}^{\text{NE}} = a_{(k)}^{\text{AF}}, \forall k$*. Otherwise,* $\theta^{\infty} > 0$ *and* $\sum_{k=1}^{K} w_k a_{(k)}^{\text{NE}} > \tau^{-1}$.

Proof See Appendix.

Corollary 4.1 *If UEs are homogeneous, i.e.* $K = 1$*, then the D2D computing participation game has a unique symmetric NE. Moreover, if* $\tau \leq \frac{1}{a^{\mathrm{AF}}}$*, then* $\theta^{\infty} = 0$ *and* $a^{NE} = a^{AF}$*; otherwise,* $\theta^{\infty} = \theta^{\dagger}$ *where* $\theta^{\dagger}$ *has the same value given in Proposition* 4.4.

Theorem 4.1 and Corollary 4.1 show that, even if UEs do not observe θ^t in real-time, the system will converge to the same state when θ^t is observed as established in Proposition 4.4. The latter case indeed can be considered as that UEs are playing best response dynamics of the population game, which converges to the predicted NE. Moreover, the thresholding effect still exhibits. If the effective infection rate is sufficiently small, then the infectious attacks extinguish. Otherwise, the infectious attacks persist.

4.6 Optimal Reward Mechanism Design

We are now ready to design the optimal reward mechanism. Under the assumption that there are K types of UEs, the operator's problem can be written as

$$\max_{r_0} \ (b_0 - r_0) \sum_{k=1}^{K} w_k (1 - \theta_{(k)}^{\infty}) a_{(k)}^{*}(\theta^{\infty}) \tag{4.10}$$

Note that θ^{∞}, $\theta_{(k)}^{\infty}$ and $a_{(k)}^{*}$ all depend on the reward mechanism r_0 even though the dependency is not explicitly expressed. Solving the above optimization problem is difficult because there are no closed-form solutions of θ^{∞}, $\theta_{(k)}^{\infty}$ and $a_{(k)}^{*}$ in terms of r_0 since they are complexly coupled as shown in the previous sections. Fortunately, our analysis shows that there is a structural property that we can exploit to solve this problem in a much easier way.

Theorem 4.2 *The optimal reward mechanism design problem* (4.10) *under infectious attacks is equivalent to the following constrained reward design problem in the attack-free network*

$$\max_{r_0} \quad (b_0 - r_0) \sum_{k=1}^{K} w_k a_{(k)}^{\mathrm{AF}}(r_0) \quad \textit{subject to} \quad \sum_{k=1}^{K} w_k a_{(k)}^{\mathrm{AF}}(r_0) \leq \tau^{-1} \tag{4.11}$$

Proof See Appendix.

Theorem 4.2 converts the reward optimization problem in the presence of infectious attack risks into an optimization problem in the attack-free network by simply adding a constraint. Because $a_{(k)}^{\mathrm{AF}}(r_0)$ can be easily computed, the converted optimization problem can be easily solved through numerical methods. In fact, since

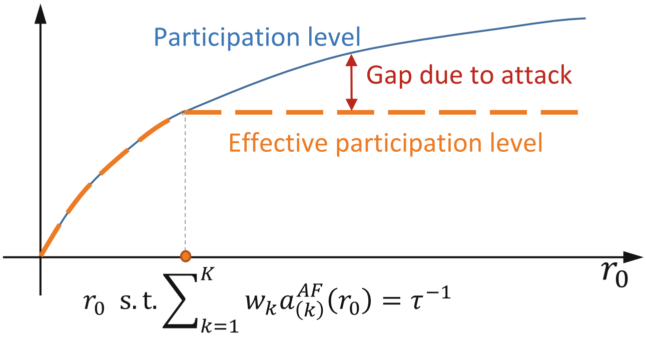


Fig. 4.5 Gap between participation level and effective participation level caused by attacks

$a_{(k)}^{\mathrm{AF}}(r_0)$ is increasing in r_0, the search space of the optimal r_0 can be restricted to $[0, \bar{r}_0]$ where $\bar{r}_0$ makes the constraint binding, i.e. $\sum_{k=1}^{K} w_k a_{(k)}^{\mathrm{AF}}(\bar{r}_0) = \tau^{-1}$.

The intuition of Theorem 4.2 is that the optimal reward mechanism must not promote too much participation that induce persistent attacks in the network. This is because too much participation does not improve the *effective* participation level due to UEs compromised by the attack (see an illustration in Fig. 4.5). Since less participation requires a smaller unit reward r_0, more utility can be obtained for the operator by employing a smaller unit reward. We call this the "less is more" phenomenon in the D2D offloading network under infectious attack risks. Corollary 4.2 further compares the optimal reward mechanisms with and without the infectious attack risks.

Corollary 4.2 *The optimal reward mechanism $r^*$for D2D offloading under infectious attack risks is no more than the optimal reward mechanism r^{AF*} in the attack-free network.*

Proof This is a direct result of Theorem 4.2. If $r^{\mathrm{AF}*} \in \mathscr{R}_2$, then $r^* = r^{\mathrm{AF}*}$. If $r^{\mathrm{AF}*} \in \mathscr{R}_1$, then $r^* < r^{\mathrm{AF}*}$.

Theorem 4.2 also implies that a larger effective infection rate reduces the operator's utility.

Corollary 4.3 *The optimal utility of the operator is non-increasing in τ.*

Proof This is a direct result of Theorem 4.2 since a larger τ imposes a more stringent constraint in the optimization problem (4.11).

Corollary 4.3 implies that the operator's utility can be improved by reducing the effective infection rate τ. This can be done by developing and deploying better security technologies that either reduce the attack success rate β or improve the recovery rate δ. Hence, the operator may consider jointly optimize the reward mechanism r_0 and security technology that results in a smaller τ. This is to solve the following optimization problem,

$$\max_{r_0,\tau} \quad (b_0 - r_0)\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}(r_0) - J(\tau) \quad \textit{subject to} \quad \sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}(r_0) \le \tau^{-1} \tag{4.12}$$

where $J(\tau)$ is the technology development cost that achieves an effective infective rate τ. Typically, $J(\tau)$ is decreasing in τ. It is evident that the optimal solution must have $\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}(r_0^*) = (\tau^*)^{-1}$. This is because if $\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}(r_0^*) < (\tau^*)^{-1}$, then we can always choose a larger $\tilde{\tau} > \tau^*$ such that $J(\tilde{\tau}) < J(\tau^*)$. This leads to a higher utility which contradicts the optimality of τ^*. Therefore, the joint optimization problem reduces to an unconstrained problem as follows

$$\max_{r_0} \quad (b_0 - r_0)\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}(r_0) - J\left(\frac{1}{\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}(r_0)}\right) \tag{4.13}$$

This problem can be easily solved using numerical methods.

4.7 Simulations

4.7.1 Simulation Setup

Since our analytical model adopts a continuous time system, we divide time into small time slots to enable the simulation. Specifically, a unit time in the continuous time system is divided into 100 slots and we simulate a large number of slots. We simulate a number $N = 100$ of mobile UEs moving in a square area of size 100×100. User mobility follows the random waypoint model. Specifically, when the UE is moving, it moves at a random speed between 0 and $v_{\max}$ per slot towards a randomly selected destination. When the UE reaches the destination, it pauses for a random number of slots between 0 and $m_{\max}$ and then selects a new destination. The parameters $v_{\max}$ and $m_{\max}$ control the mobility level of the network and will be varied to study the sensitivity of the random server-requester matching approximation to the actual UE mobility and server-requester matching.

In every slot, with probability p a UE has computation tasks to offload and hence becomes a requester. When a UE is not a requester, it is able to provide D2D offloading service. Therefore, with probability $1-p$ the UE is a potential server. The number of tasks of each requester in every slot is randomly selected between 1 and W_{max}. We assume requesters have a minimum transmission rate requirement $r^{\mathrm{tx}}_{\min}$ to guarantee the worst-case transmission delay. Suppose UEs operate at a fixed transmission power p^{tx}_i, then the transmission rate between two UEs i and j is given by the Shannon capacity:

$$r^{\mathrm{tx}}_{ij} = W \log_2\left(1 + \frac{p^{\mathrm{tx}}_i H_{ij}}{\sigma^2}\right) \tag{4.14}$$

where W is the channel bandwidth, H_{ij} is the channel condition, and σ^2 is the noise power. Therefore, for a UE j to be a potential server of a requester UE i, it must satisfy $r_{ij}^{\text{tx}} \geq r_{\min}^{\text{tx}}$. With fixed channel bandwidth and transmission power, the transmission rates mainly depend on the channel conditions which are modeled by free space pathloss with slow fading in the simulation. Specifically, we set the minimum transmission rate requirement $r_{\min}^{\text{tx}} = 10$ Mps, channel bandwidth $W = 180$ kHz, $p_i^{\text{tx}} = 10$ dBm, and noise power $\sigma^2 = -174$ dBm/Hz. Suppose that UEs are obedient, then the operator ranks the potential server UEs for each requester according to the transmission rates and assigns tasks sequentially to the server UEs with highest transmission rate until all the tasks have been assigned. The probability η that a potential server UE receives an offloading request in the obedient UE case can be estimated in simulations. Estimating this probability is very important for the conversion of the participation level in continuous time into its counterpart in discrete time. Specifically, a chosen participation level a per unit time in continuous time is converted to a participation probability $\frac{a}{100(1-p)\eta}$ in each slot when the UE is a potential server in discrete time. With this conversion, in the strategic UE case, the operator assigns one task to one of the potential server UE with probability $\frac{a}{100(1-p)\eta}$.

4.7.2 System Dynamics

Figure 4.6 illustrates the system dynamics for non-strategic UEs who follow a prescribed participation strategy. Two types of UEs are considered in this simulation. Type 1 UEs adopt a participation level $a_{(1)} = 3$ and Type 2 UEs adopt a participation level $a_{(2)} = 5$. The fractions of these two types of users are $w_{(1)} = 0.3$ and $w_{(2)} = 0.7$, respectively. Therefore, the predicted critical effective infection rate $\tau_c = 0.227$ according to Proposition 4.5. We fix $\delta = 1$ and show the results for $\beta = 0.2$ and $\beta = 0.4$, which correspond to $\tau = 0.2$ and $\tau = 0.4$, respectively. As shown in Fig. 4.6,

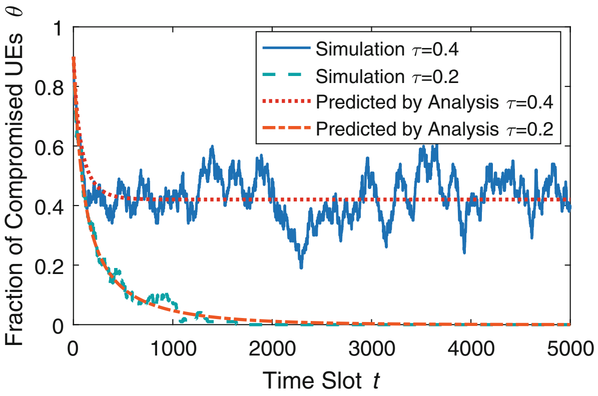


Fig. 4.6 Epidemic dynamics for non-strategic UE

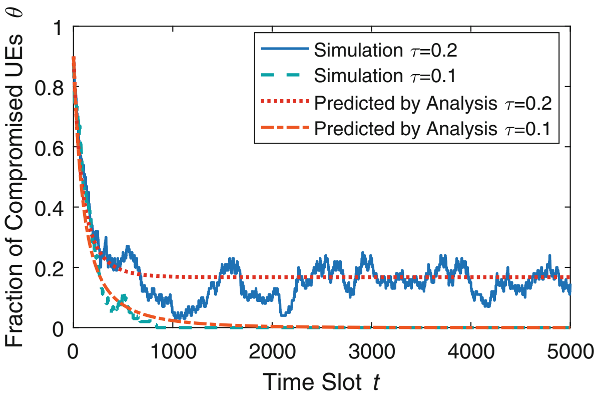


Fig. 4.7 Epidemic dynamics for strategic UEs

when $\tau < \tau_c$, the infections extinguish over time. When $\tau > \tau_c$, infections persist in the system at a compromise level around 0.42. Because we used a relatively small finite UE population in the simulation, there are still fluctuations in the results. However, the predicted dynamics by our analysis is in accordance to the simulation results and the fluctuations will be less with a larger UE population.

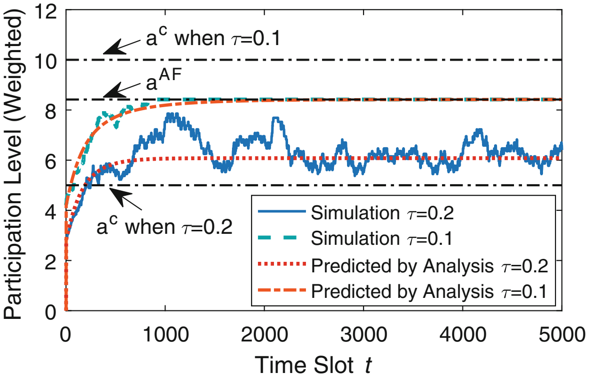
Figure 4.7 illustrates the system dynamics for strategic UEs for the same setting as above. The difference is that, since UEs are strategic, they will decide their participation levels by themselves. The user evaluation functions are chosen as $v_{(1)}(x) = \sqrt{x}$ and $v_{(2)} = 1.5\sqrt{x}$. The costs for users are the same $c = 0.35$. The reward offered by the operator is set as $r = 2.2$. Therefore, the critical infection rate is computed to be $\tau_c = 0.12$ according to Theorem 4.1. As shown in Fig. 4.7, infections extinguish over time when $\tau = 0.1$ which is smaller than τ_c and persist when $\tau = 0.2$ which is larger than τ_c. Notice that in the non-strategic UE case, $\tau = 0.2$ will instead make attacks extinguish.

To better understand what is happening behind this epidemic dynamics, we show the evolution of UE participation levels (weighted average of the two types) in Fig. 4.8. When the effective infection rate is lower, UEs tend to choose a higher participation level. Regardless of the exact value of τ, the converged participation level does not exceed the optimal participation level a^{AF} in the attack-free network (which does not depend on τ). When $\tau = 0.2$, the converged value is greater than the corresponding critical participation level a^c (which depends on τ) and hence, the attacks persist. When $\tau = 0.1$, the converged value is smaller than the corresponding critical participation level a^c, thereby eliminating the attacks.

4.7.3 Optimal Reward Mechanism

Now we consider the operator's utility maximization problem. In this set of simulations, the benefit for the operator is set as $b_0 = 6$. Figure 4.9 shows the impact

Fig. 4.8 Evolution of UE participation levels



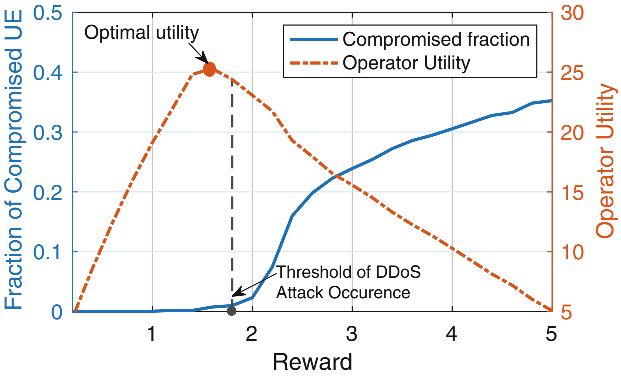


Fig. 4.9 Impact of reward on system compromise state and the operator’s utility

of the reward on the fraction of compromised UEs in the network as well as the operator’s utility. As the reward increases, UEs have more incentives to participate and when the reward increases to a certain point, infections become persistent. As a result, further increasing the reward r_0 decreases the operator’s utility since the effective participation level of UEs does not improve as shown in Fig. 4.10. This is the predicted “Less is More” phenomenon. This result is important for the operator to determine the optimal reward mechanism that is security-aware. Figures 4.11 and 4.12 further show the operator’s utility and the fraction of compromised UEs as functions of the reward r_0 and the effective infection rate τ, which are in accordance with our analysis.

4.7.4 Performance Comparison

We compare the operator utility achieved by our security-aware design with two baseline approaches: (1) Fully cooperative: all UEs are assumed to be obedient and

Fig. 4.10 Impact of reward on effective participation level

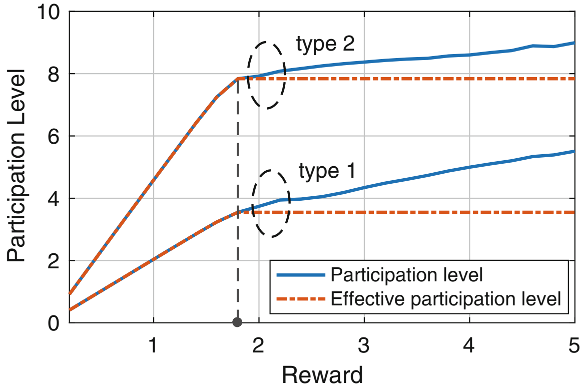


Fig. 4.11 Operator's utility as a function of r_0 and τ

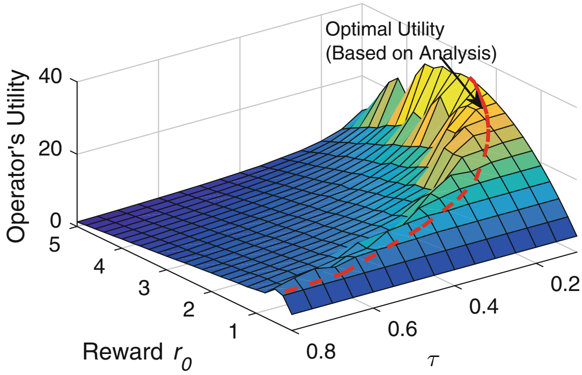
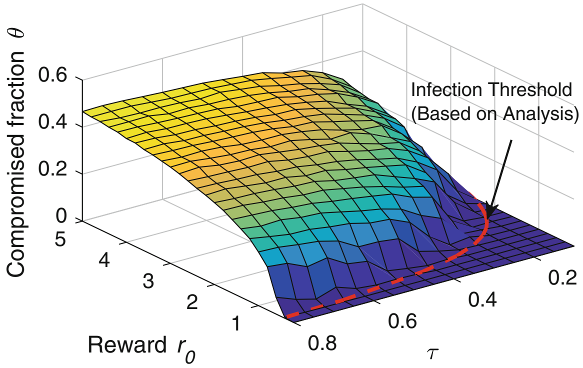
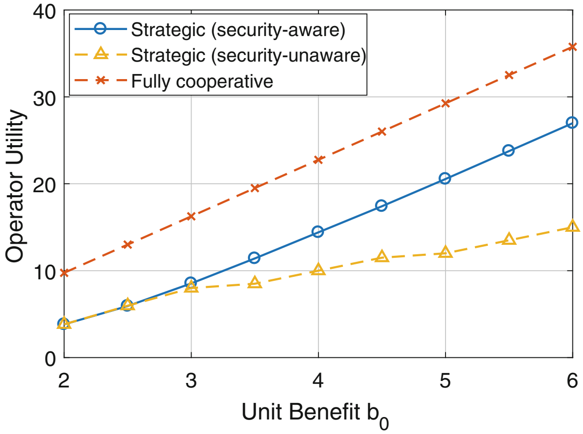


Fig. 4.12 System compromise state as a function of r_0 and τ



participate in the D2D offloading system in a fully cooperative manner. In this case, the operator does not need to pay rewards to the UEs. (2) Security-unaware: UEs are assumed to be self-interested. However, both the UEs and the operator do not take into account the potential security risk in participating in D2D offloading when making their decisions. Figure 4.13 illustrates the result by varying b_0. As expected,

Fig. 4.13 Performance comparison



the achievable utility is the highest in the fully cooperative case, and the security-aware design outperforms the security-unaware design. When the unit benefit b_0 is small, the security-aware design and the security-unaware design achieve the same utility. This is because when b_0 is small, the operator is only willing to provide a low reward r_0 to promote D2D offloading, in which case the induced low participation level of UEs does not cause persistent security risks even when the operator and the UEs are not security-aware.

4.7.5 Impact of UE Mobility

Finally, we investigate the impact of UE mobility on the accuracy of our model and analysis by varying the moving speed of UEs. Figures 4.14 and 4.15 show how the UE participation levels and the fraction of compromised UEs change with UE mobility, respectively. In our model, we assumed that a UE receives requests from other UEs uniformly randomly, which is a good approximation when UEs' mobility is fast. However, when UEs' mobility is slow, they will more likely have localized interactions with only a subset of UEs with high probability. For instance, in practice, people are more likely to appear in the same locations at the same time with their family, friends and colleagues. As shown in Figs. 4.14 and 4.15, when UE mobility is fast, our analytical results are very well aligned with the simulation results. However, when UE mobility is slow, there is an obvious deviation from our analysis, suggesting that new models are needed to handle low mobility network scenarios. This is an interesting future work direction.

Fig. 4.14 Impact of mobility on the participation levels

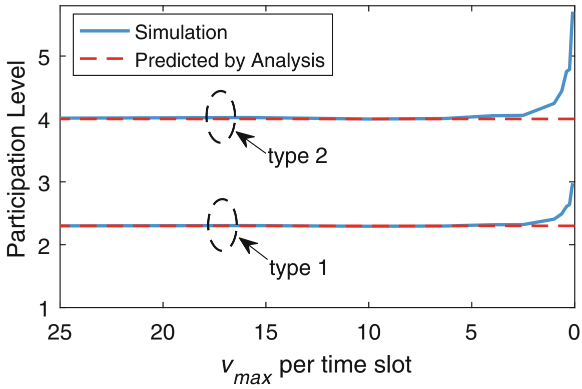
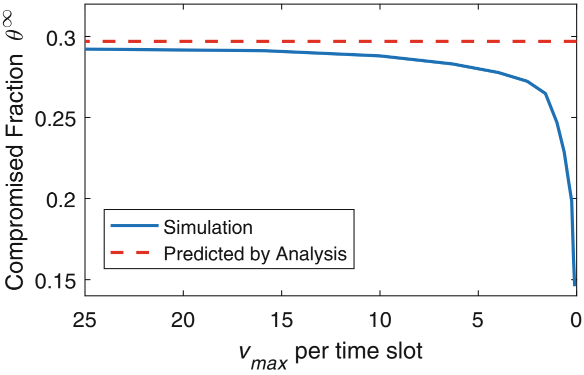


Fig. 4.15 Impact of mobility on the system compromise state



4.8 Conclusions

In this chapter, we investigated the important but much less studied incentive mechanism design problem in dynamic networks where users' incentives and security risks they face are intricately coupled. We adopted a dynamic non-cooperative game theoretic approach to understand how user collaboration incentives are influenced by interdependent security risks such as the infectious attack risks, and how the attack risks evolve, propagate, persist and extinguish depending on users' strategic choices. This understanding allowed us to develop security-aware incentive mechanisms that are able to combat and mitigate attacks in D2D offloading systems. Our study leverages the classic epidemic models, but on the other hand, it represents a significant departure from these models since users are strategically choosing their actions rather than obediently following certain prescribed rules. Our model and analysis not only provide new and important insights and guidelines for designing more efficient and more secure D2D offloading networks but also can be adapted to solve many other challenging problems in other cooperative networks where users face interdependent security risks. Future work includes investigating user interaction models that are more localized and social network-based, and user heterogeneity in terms of adopted security technologies.

Appendix

Proof of Proposition 4.1 We omit the UE index i in the subscript of $v_i(\cdot)$, c_i and q_i for brevity. Given the foresighted utility in (4.8), determining the optimal participation level boils down to investigating the first-order condition of (4.8). The derivative of $U(a_i, \theta)$ is

$$U'(a_i, \theta) = (\rho + \delta)\frac{u'(a_i)(\rho + \delta + \beta\theta a_i) - u(a_i)\beta\theta - \beta\theta q}{(\rho + \delta + \beta\theta a_i)^2} \tag{4.15}$$

For brevity, let $f(a_i) = u'(a_i)(\rho + \delta + \beta\theta a_i) - u(a_i)\beta\theta - \beta\theta q$, which has the same sign of $U'(a_i, \theta)$. First, we have

$$f'(a_i) = u''(a_i)(\rho + \delta + \beta\theta a_i) = r_0^2 v''(r_0 a_i)(\rho + \delta + \beta\theta a_i) < 0 \tag{4.16}$$

Therefore, $f(a_i)$ is monotonically decreasing. Next, we investigate the signs of $f(M)$ and $f(0)$.

$$f(M) = u'(M)(\rho + \delta + \beta\theta M) - u(M)\beta\theta - \beta\theta q < 0 \tag{4.17}$$

The inequality is because $u'(M) < 0$ and $u(M) > 0$ according to Assumption 1(2). Also,

$$f(0) = u'(0)(\rho + \delta) - u(0)\beta\theta - \beta\theta q = (r_0 v'(0) - c)(\rho + \delta) - \beta\theta q \tag{4.18}$$

Therefore, if $\theta < \frac{(r_0 v'(0) - c)(\rho + \delta)}{q\beta}$, then $f(0) > 0$. Otherwise, $f(0) \leq 0$. This means that if $\theta \geq \frac{(r_0 v'(0) - c)(\rho + \delta)}{q\beta}$, then the optimal $a^* = 0$ and otherwise, there exists an optimal participation level $a^* > 0$, which is the unique solution of

$$u'(a_i)(\rho + \delta\beta\theta a_i) - u(a_i)\beta\theta - \beta\theta q = 0 \tag{4.19}$$

To investigate the monotonicity of a_i^* with θ, we rewrite the above equation as follows

$$\frac{u(a_i) + q}{u'(a_i)} - a_i = \frac{\rho + \delta}{\beta\theta} \tag{4.20}$$

Notice that $u(a_i)$ does not have ρ, δ, β or θ in its expression according to (4.2). The first-order derivative of the left-hand side function of a_i is

$$\frac{u'(a_i)u'(a_i) - (u(a_i)+q)u''(a_i)}{(u'(a_i))^2} - 1 = \frac{-(u(a_i)+q)u''(a_i)}{(u'(a_i))^2} > 0 \tag{4.21}$$

The last inequality is because $u(a) > 0$, $\forall a \in [0, M]$ and $u''(a) = r_0^2 v''(r_0 a_i) < 0$. Therefore the left-hand side of (4.20) is monotonically increasing in a_i. Since the right-hand-side is decreasing in θ, a_i decreases with the increase of θ.

Proof of Proposition 4.3 Consider the compromise state dynamics given a symmetric strategy a. For any θ, the change in θ in a small interval dt is $d\theta = -\theta\delta dt + (1-\theta)\theta\beta a dt = \theta((1-\theta)\beta a - \delta)dt$. Clearly, if $\tau > \frac{1}{a} \triangleq \tau_c$, then for any $\theta > 1 - \frac{\delta}{\beta a} \triangleq \theta^*$, $d\theta < 0$ and for $\theta < \theta*$, $d\theta > 0$. Therefore, the dynamic system must converge to $\theta*$. If $\tau \leq \tau_c$, then for any $\theta > 0$, $d\theta < 0$. This means that the dynamic system converges to 0.

Proof of Proposition 4.4 The system dynamics can be rewritten as

$$d\theta^t = \theta^t[(1-\theta^t)\beta a^*(\theta^t) - \delta]dt \tag{4.22}$$

We are interested in the sign of $d\theta^t$ for different values of θ^t. Since $\theta^t \geq 0$, what matters is the sign of $f(\theta^t) \triangleq (1-\theta^t)\beta a^*(\theta^t) - \delta$. For $\theta^t \geq \bar{\theta}$, $a*(\theta^t) = 0$ and hence, $f(\theta^t) = -\delta < 0$. For $\theta^t < \bar{\theta}, f(\theta^t)$ is decreasing in θ^t because $a*(\theta^t)$ is decreasing in θ^t according to Proposition 4.1. Now consider the sign of $f(0) = \beta a*(0) - \delta$. According to (4.19), $a*(0)$ is the solution to $u'(a) = 0$, which is the same as the optimal participation level in the attack-free case. Specifically, $a*(0) = a^{\mathrm{AF}}$. If $\beta a^{\mathrm{AF}} - \delta < 0$, then $f(\theta^t) < 0$ for all θ^t. Therefore, the system converges to $\theta^\infty = 0$. If $\beta a^{\mathrm{AF}} - \delta \geq 0$, then there exists a unique point $\theta^\dagger \in [0, \bar{\theta})$ such that for $\theta^t > \theta^\dagger, f(\theta^t) < 0$ and for $\theta^t < \theta^\dagger, f(\theta^t) > 0$. This means that the system compromise state will converge to θ^t. Moreover, θ^t is the solution of $(1-\theta)\beta a*(\theta) - \delta = 0$.

Proof of Proposition 4.5 Let $\theta_{(k)}$ be the fraction of compromised UEs among all type k UEs. In the steady state, we have the following relation

$$\theta_{(k)}^\infty = \frac{\delta^{-1}}{\delta^{-1} + (\theta^\infty \beta a_{(k)})^{-1}} = \frac{\tau\theta^\infty a_{(k)}}{\tau\theta^\infty a_{(k)} + 1} \tag{4.23}$$

where the fraction of the compromised UEs among all UEs is $\theta^\infty = \sum_k w_k \theta_{(k)}^\infty$. It is clear from the above equation that if $\theta^\infty = 0$, then $\theta_{(k)}^\infty = 0$, $\forall k$. Hence, $\theta^\infty = 0$ is a trivial solution in which $a_{(k)}$, $\forall k$ can be any value. We now study the non-trivial solution $\theta^\infty > 0$. Rearranging the above equation, we have $\theta_{(k)}^\infty = (1-\theta_{(k)}^\infty)\theta^\infty \tau a_{(k)}$. Summing up over k and multiplying by w_k on both sides, we have

$$\theta^\infty = \sum_{k=1}^{K} w_k \theta_{(k)}^\infty = \tau\theta^\infty \sum_{k=1}^{K} w_k (1-\theta_{(k)}^\infty) a_{(k)} \tag{4.24}$$

This leads to

$$\tau \sum_{k=1}^{K} w_k (1 - \theta_{(k)}^{\infty}) a_{(k)} = 1 \tag{4.25}$$

If $\tau \le \tau_c = \frac{1}{\sum_{k=1}^{K} w_k a_{(k)}}$, then clearly there is no non-trivial solution of $\theta_{(k)}^{\infty}$ of the above equation. This implies that the only solution is $\theta^{\infty} = \theta_{(k)}^{\infty} = 0, \forall k$, which proves the first half of this proposition. Next, we show that if $\tau > \tau_c$, there indeed exists a unique solution $\theta^{\infty} > 0$. Substituting (4.23) into (4.25) yields

$$\sum_{k=1}^{K} \frac{\tau w_k a_{(k)}}{\tau \theta^{\infty} a_{(k)} + 1} = 1 \tag{4.26}$$

Clearly the left-hand side of the above equation LHS(θ^{∞}) is decreasing in θ^{∞}. Moreover LHS(0) $= \tau \sum_{k=1}^{K} w_k a_{(k)} > 1$, and LHS(1) $= \tau \sum_{k=1}^{K} \frac{w_k a_{(k)}}{\tau a_{(k)} + 1} < \sum_{k=1}^{K} w_k = 1$. Therefore, there is a unique solution of $\theta^{\infty} \in (0, 1)$.

Proof of Theorem 4.1 Consider type k UEs. Each UE chooses the individual optimal participation level determined by the following equation

$$u'_{(k)}(a_i)(\rho + \delta\beta\theta^{\infty} a_i) - u_{(k)}(a_i)\beta\theta^{\infty} - \beta\theta^{\infty} q_{(k)} = 0 \tag{4.27}$$

Given the same β, δ, θ^{∞}, there is a unique optimal solution a_i^* according to Proposition 4.1. Therefore, if an equilibrium exists, UEs of the same type must choose the same participation level. To prove the existence of NE is to prove that the following function has a fixed point θ^{∞} based on our analysis in the proof of Proposition 4.5:

$$\theta^{\infty} = \tau\theta^{\infty} \sum_{k=1}^{K} w_k (1 - \theta_{(k)}^{\infty}) a_{(k)}(\theta^{\infty}) \tag{4.28}$$

Note that the difference from (4.24) is that $a_{(k)}(\theta^{\infty})$ is a function of θ^{∞} rather than a prescribed action.

First, we investigate if $\theta^{\infty} = 0$ could be a fixed point. If $\theta^{\infty} = 0$, then $a_{(k)}^* = a_{(k)}^{\mathrm{AF}}, \forall k$, which is the optimal participation level in the attack-free network. Therefore, if $\tau \le \frac{1}{\sum_{k=1}^{K} w_k a_{(k)}^{\mathrm{AF}}}$, then $\theta^{\infty} = 0$ is a fixed point. Otherwise, $\theta^{\infty} = 0$ is not a fixed point.

Next, we investigate if there is any $\theta^{\infty} > 0$ that can be fixed point. This is to show, according to (4.28), if there is a solution to

$$\sum_{k=1}^{K} \frac{w_k \tau a_{(k)}(\theta^{\infty})}{\tau\theta^{\infty} a_{(k)}(\theta^{\infty}) + 1} = 1 \tag{4.29}$$

Denote the left-hand side function by $f(\theta^{\infty})$.

$$f'(\theta^{\infty}) = \sum_{k=1}^{K} w_k \tau \frac{a'_{(k)}(\theta^{\infty}) - \tau a^2_{(k)}(\theta^{\infty})}{(\tau\theta^{\infty} a^*_{(k)}(\theta^{\infty}) + 1)^2} < 0 \tag{4.30}$$

The inequality is because $a_{(k)}(\theta^{\infty})$ decreases with θ^{∞} according to Proposition 4.1. Moreover, $f(1) < \sum_{k=1}^{K} w_k = 1$ and $f(0) = \tau \sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}$. Therefore, if $\tau > \frac{1}{\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)}}$, then $f(0) > 1$. This means that there exists a unique positive solution θ^{∞}.

Proof of Theorem 4.2 We divide the reward mechanism r_0 into two categories $\mathscr{R}_1$ and $\mathscr{R}_2$. Consider any reward mechanism r_0, if the resulting $\sum_{k=1}^{K} a^*_{(k)}(\theta^{\infty}) \geq \tau^{-1}$, then $r_0 \in \mathscr{R}_1$. Otherwise $r_0 \in \mathscr{R}_2$.

Now, according to Theorem 4.1, if $r_0 \in \mathscr{R}_1$, then we also have $\sum_{k=1}^{K} w_{(k)}(1 - \theta^{\infty})a^*_{(k)}(\theta^{\infty}) = \tau^{-1}$, which is a constant that does not depend on the exact value of r_0. Therefore, the optimal r_0 in $\mathscr{R}_1$ must be the smallest possible r_0 in order to maximize the operator's utility. The smallest r_0 is the one such that $\sum_{k=1}^{K} w_k a^*_{(k)}(\theta^{\infty}) = \tau^{-1}$ and $\theta^{\infty} = 0$. Since $\theta^{\infty} = 0$, $\sum_{k=1}^{K} w_k a^*_{(k)}(\theta^{\infty}) = \tau^{-1}$ is equivalent to $\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)} = \tau^{-1}$. This means that if $r_0 \in \mathscr{R}_1$ is the optimal solution, it is also a feasible solution of the above constrained optimization problem.

If $r_0 \in \mathscr{R}_2$, then according to Theorem 4.1, we have $\theta^{\infty} = 0$. Again, since $\theta^{\infty} = 0$, $\sum_{k=1}^{K} a^*_{(k)}(\theta^{\infty}) < \tau^{-1}$ is equivalent to $\sum_{k=1}^{K} w_k a^{\mathrm{AF}}_{(k)} < \tau^{-1}$. This also proves that if $r_0 \in \mathscr{R}_2$ is the optimal solution, it is also a feasible solution of the above constrained optimization problem.

References

1. H.T. Dinh, C. Lee, D. Niyato, P. Wang, A survey of mobile cloud computing: architecture, applications, and approaches. Wirel. Commun. Mob. Comput. **13**(18), 1587–1611 (2013)
2. Y. Mao, C. You, J. Zhang, K. Huang, K.B. Letaief, A survey on mobile edge computing: the communication perspective. IEEE Commun. Surv. Tutorials **19**(4), 2322–2358 (2017)
3. M. Chiang, T. Zhang, Fog and IoT: an overview of research opportunities. IEEE Internet Things J. **3**(6), 854–864 (2016)
4. E.E. Marinelli, Hyrax: cloud computing on mobile devices using MapReduce. DTIC Document, Technical Report (2009)
5. C. Shi, V. Lakafosis, M.H. Ammar, E.W. Zegura, Serendipity: enabling remote computing among intermittently connected mobile devices, in *ACM International Symposium on Mobile Ad Hoc Networking and Computing* (ACM, New York, 2012), pp. 145–154
6. G. Huerta-Canepa, D. Lee, A virtual cloud computing provider for mobile devices, in *Proceedings of the 1st ACM Workshop on Mobile Cloud Computing and Services: Social Networks and Beyond* (ACM, New York, 2010), p. 6
7. Y. Li, W. Wang, Can mobile cloudlets support mobile applications?, in *IEEE Conference on Computer Communications (INFOCOM)*, April 2014, pp. 1060–1068
8. M. Haus, M. Waqas, A.Y. Ding, Y. Li, S. Tarkoma, J. Ott, Security and privacy in device-to-device (D2D) communication: a review. IEEE Commun. Surv. Tutorials **19**(2), 1054–1079 (2017)

9. A. Mtibaa, K. Harras, H. Alnuweiri, Friend or foe? Detecting and isolating malicious nodes in mobile edge computing platforms, in *IEEE International Conference on Cloud Computing Technology and Science* (IEEE, Piscataway, 2015), pp. 42–49
10. Z. Lu, W. Wang, C. Wang, How can botnets cause storms? Understanding the evolution and impact of mobile botnets, in *IEEE International Conference on Computer Communications (INFOCOM)* (IEEE, Piscataway, 2014), pp. 1501–1509
11. D. Fudenberg, J. Tirole, *Game Theory*, vol. 393 (MIT Press, Cambridge, 1991)
12. W.O. Kermack, A.G. McKendrick, A contribution to the mathematical theory of epidemics. Proc. R. Soc. Lond. A Math. Phys. Eng. Sci. **115**(772), 700–721 (1927)
13. N. Fernando, S.W. Loke, W. Rahayu, Mobile cloud computing: a survey. Futur. Gener. Comput. Syst. **29**(1), 84–106 (2013)
14. D. Chatzopoulos, M. Ahmadi, S. Kosta, P. Hui, Have you asked your neighbors? A hidden market approach for device-to-device offloading, in *IEEE International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (IEEE, Piscataway, 2016), pp. 1–9
15. D. Huang, P. Wang, D. Niyato, A dynamic offloading algorithm for mobile computing. IEEE Trans. Wirel. Commun. **11**(6), 1991–1995 (2012)
16. M. Satyanarayanan, P. Bahl, R. Caceres, N. Davies, The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comput. **8**(4), 14–23 (2009)
17. Z. Chang, S. Zhou, T. Ristaniemi, Z. Niu, Collaborative mobile clouds: an energy efficient paradigm for content sharing. IEEE Wirel. Commun. **25**(2), 186–192 (2017)
18. M. Chen, Y. Hao, Y. Li, C.F. Lai, D. Wu, On the computation offloading at ad hoc cloudlet: architecture and service modes. IEEE Commun. Mag. **53**(6), 18–24 (2015)
19. A. Asadi, Q. Wang, V. Mancuso, A survey on device-to-device communication in cellular networks. IEEE Commun. Surv. Tutorials **16**(4), 1801–1819 (2014)
20. X. Wu, S. Tavildar, S. Shakkottai, T. Richardson, J. Li, R. Laroia, A. Jovicic, FlashLinQ: a synchronous distributed scheduler for peer-to-peer ad hoc networks. IEEE/ACM Trans. Netw. **21**(4), 1215–1228 (2013)
21. X. Lin, J.G. Andrews, A. Ghosh, R. Ratasuk, An overview of 3GPP device-to-device proximity services. IEEE Commun. Mag. **52**(4), 40–48 (2014)
22. Y. Li, L. Sun, W. Wang, Exploring device-to-device communication for mobile cloud computing, in *IEEE International Conference on Communications (ICC)* (IEEE, Piscataway, 2014), pp. 2239–2244
23. P. Li, S. Guo, Incentive mechanisms for device-to-device communications. IEEE Netw. **29**(4), 75–79 (2015)
24. J. Xu, M. Van Der Schaar, Token system design for autonomic wireless relay networks. IEEE Trans. Commun. **61**(7), 2924–2935 (2013)
25. N. Mastronarde, V. Patel, J. Xu, L. Liu, M. van der Schaar, To relay or not to relay: learning device-to-device relaying strategies in cellular networks. IEEE Trans. Mob. Comput. **15**(6), 1569–1585 (2016)
26. Y. Zhang, L. Song, W. Saad, Z. Dawy, Z. Han, Contract-based incentive mechanisms for device-to-device communications in cellular networks. IEEE J. Sel. Areas Commun. **33**(10), 2144–2155 (2015)
27. J.O. Kephart, S.R. White, Directed-graph epidemiological models of computer viruses, in *IEEE Computer Society Symposium on Research in Security and Privacy* (IEEE, Piscataway, 1991), pp. 343–359
28. Y. Wang, D. Chakrabarti, C. Wang, C. Faloutsos, Epidemic spreading in real networks: an eigenvalue viewpoint, in *22nd International Symposium on Reliable Distributed Systems* (IEEE, Piscataway, 2003), pp. 25–34
29. N.T. Bailey et al., *The Mathematical Theory of Infectious Diseases and Its Applications* (Charles Griffin and Company Ltd, Bucks, 1975)
30. P. Van Mieghem, J. Omic, R. Kooij, Virus spread in networks. IEEE/ACM Trans. Netw. **17**(1), 1–14 (2009)
31. R. Yin, C. Zhong, G. Yu, Z. Zhang, K. K. Wong, X. Chen, Joint spectrum and power allocation for D2D communications underlaying cellular networks. IEEE Trans. Veh. Technol. **65**(4), 2182–2195 (2016)

32. F. Wang, L. Song, Z. Han, Q. Zhao, X. Wang, Joint scheduling and resource allocation for device-to-device underlay communication, in *2013 IEEE Wireless Communications and Networking Conference (WCNC)* (IEEE, Piscataway, 2013) pp. 134–139
33. Wi-Fi Direct, http://www.wi-fi.org/discoverwi-fi/wi-fi-direct
34. LTE Direct, https://www.qualcomm.com/invention/research/projects/lte-direct
35. ATT Data Perks, https://www.att.com/att/dataperks/en/index.html
36. G.J. Mailath, L. Samuelson, *Repeated Games and Reputations: Long-Run Relationships* (Oxford University Press, Oxford, 2006)
37. K. Doya, Reinforcement learning in continuous time and space. Neural Comput. **12**(1), 219–245 (2000)

Chapter 5
Enhance Physical Layer Security via Channel Randomization with Reconfigurable Antennas

Yanjun Pan, Ming Li, Yantian Hou, Ryan M. Gerdes, and Bedri A. Cetiner

Abstract Secure wireless communication techniques based on physical (PHY) layer properties are promising alternatives or complements to traditional upper-layer cryptography-based solutions, due to the capability of achieving message confidentiality or integrity and authentication protection without pre-shared secrets. While many theoretical results are available, there are few practical PHY-layer security schemes, mainly because the requirement of channel advantage between the legitimate users versus the attacker's is hard to satisfy in all cases. Recent research shows that channel randomization, which proactively and dynamically perturbs the physical channel so as to create an artificial channel advantage, is helpful to enhance certain PHY-layer security goals such as secrecy. However, a systematic study of the foundations of such an approach and its applicability is needed. In this chapter, we first survey the state-of-the-art in PHY-layer security and identify their main limitations as well as challenges. Then we examine the principles of channel randomization and explore its application to achieve in-band message integrity and authentication. Especially, we focus on preventing active signal manipulation attacks and use reconfigurable antennas to systematically randomize the channel

Y. Pan · M. Li (✉)
The University of Arizona, Tucson, AZ, USA
e-mail: lim@email.arizona.edu

Y. Hou
Boise State University, Boise, ID, USA
e-mail: yantianhou@boisestate.edu

R. M. Gerdes
Virginia Tech, Arlington, VA, USA
e-mail: rgerdes@vt.edu

B. A. Cetiner
Utah State University, Logan, UT, USA
e-mail: bedri.cetiner@usu.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances
in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_5

such that it is unpredictable to the active attacker. Both theoretical and experimental results show that it is a feasible and effective approach. Other applications and future directions are discussed in the end.

5.1 Introduction

Wireless communication technology has been widely deployed for an increasingly large number of applications such as WiFi and Internet-of-Things. However, the broadcast nature of the wireless channel poses numerous security challenges, ranging from eavesdropping sensitive information, malicious jamming the communication to modify critical control messages via signal manipulation. Typically, the first step toward securing the communications between transceivers is trust establishment, which includes device authentication and key agreement. The former is needed to verify the communicating devices' identities, whereas the latter establishes a secure (private) channel over a public medium. The prevailing methods for trust establishment either involve the manual input of a secret (e.g., a password or a PIN) to each device, or by preloading devices with some default secret. However, key preloading solutions pose significant scalability, usability, and interoperability challenges. Many new wireless devices lack the necessary interfaces to enter or change passwords. Even if those passwords are entered a prior, manufacturers frequently opt for default secrets that are easily leaked. Alternative solutions relying on public key cryptosystems require a public key infrastructure, which is not yet deployed at large scale. In addition, key revocation is very challenging with public key infrastructures, since frequent reach-back to a central server may not be feasible due to the intermittent and ad hoc nature of the network connections, especially for applications relevant to remote military deployment or disaster response.

Thus, an important research objective is to establish secure wireless communications in the absence of preloaded secrets. Ideally, this should be achieved only using in-band communications (between devices with a common radio interface), due to interoperability and usability requirements. In the past, physical (PHY)-layer security has been proposed as a promising means to protect the security of wireless communications under the information-theoretic security notion, without any pre-shared secrets. For instance, many PHY-layer characteristics based key agreement schemes have been proposed in the literature [24, 33, 49]. However, we emphasize that existing PHY-layer security approaches mostly aim at achieving confidentiality (i.e., preventing eavesdropping against a passive adversary) and do not consider active attacks. Among them, Man-in-the-Middle (MitM) attacks through advanced signal manipulations such as signal injection and cancellation are especially difficult to detect and prevent, due to lack of authentication without any pre-shared secrets. For example, Eberz et al. demonstrated a practical MitM attack against existing received signal strength based PHY-layer key agreement schemes [14], where an active attacker inferred the secret key of legitimate parties by intelligently injecting its own messages. Besides, the more advanced signal cancellation attacks which aim at completely canceling out the received signal at the receiver's side have shown to be feasible in recent years [16, 17, 22, 39]. We note that signal cancellation attacks are more powerful than traditional active attacks such as signal injection, overshadowing, or jamming; however, few effective defenses against them are known to date.

In this chapter, we examine the principles of channel randomization and explore its application to achieve in-band message integrity and authentication. Especially, we focus on preventing active signal manipulations and use reconfigurable antennas to systematically randomize the channel. The general idea of channel randomization was recently adopted in the PHY-layer security context to increase the randomness of the wireless channel for higher secrecy rates. Here we further adapt this approach to proactively and dynamically perturb the physical channel so as to create an artificial advantage against the attacker, which can be viewed as one of the proactive/dynamic defense (or moving target defense) mechanisms. In general, moving target defense techniques are defense mechanisms via changing system characteristics to increase uncertainty and complexity for attackers [11]. For example, IP-hopping [28], in which the transparency is achieved by keeping the real host's IP address and associating each host with a virtual random IP address, was used to change the host's IP address, thereby increasing the complexity of the network seen by the attacker. In fact, the basic idea of our channel randomization approach is to regard the channel state information (CSI) as a partial secret of the legitimate communicating pairs. By leveraging the state diversity and fast reconfigurability of reconfigurable antennas, we can proactively randomize CSI frequently and thwart attackers from accurately estimating or predicting it (and generating the desired waveform to cancel the legitimate signal).

The rest of this chapter is organized as follows. Section 5.2 presents the state-of-the-art PHY-layer security schemes. Section 5.3. introduces the system and attack models, as well as the game-theoretical framework that analyzes the attacker/defender's strategies and their optimal utilities. Section 5.4 gives our channel randomization approach and introduces our method to protect message integrity in practice. In Sect. 5.5, we present the experimental study and performance analysis of the system. Section 5.6. concludes the chapter and points out some future research directions.

5.2 State-of-the-Art Physical Layer Secure Communication Schemes

In this section, we provide a survey on the existing PHY-layer schemes for confidentiality, integrity and authentication services in wireless communications. Various PHY-layer security techniques are reviewed and compared, including information-theoretic schemes and PHY-layer secret key generation methods for message confidentiality, followed by out-of-band and in-band approaches for message integrity and authentication. The main limitations and challenges for existing channel randomization approaches are also investigated.

5.2.1 Message Confidentiality Protection

The exploration of PHY-layer security was pioneered by Wyner's work on the wiretap channel [47], where the eavesdropper's channel is assumed to be a degraded version of transceivers'. The main idea is to employ the so called *channel advantage* to achieve secret and reliable communications between two legitimate transceivers in the presence of an eavesdropper, as long as their channel is better than the eavesdropper's channel. The rate of the secret communications is characterized by the channel's secrecy capacity and this result was later extended to a basic Gaussian channel that better models wireless communication systems [31]. Notably, their schemes make no assumption on shared keys between transceivers, which makes it quite attractive for various security purposes such as key agreement. Although Wyner's idea has drawn significant attention by the security community (e.g., [13, 30]), it has remained largely impractical due to the requirement of a better channel for the legitimate receiver than the eavesdropper.

Later, Maurer proposed the idea of common randomness, in which two parties can both tune to a common radio signal source and extract a secret key from it [34], provided that such common signal source is not error free for either the legitimate parties or the adversary. Existing works adopted various types of common randomness from the radio channel including the received signal strength (RSS) of a fast-fading reciprocal communication channel [24, 33, 49] and channel state information (CSI) phase information [46].

Since the natural wireless channel may not contain enough randomness to satisfy high key generation rates, other works have explored the cooperative/friendly jamming to increase the randomness or the advantage of the wireless channel. For example, Anand et al. [3] proposed orthogonal blinding/masked beamforming, where transmitters protect messages by sending artificial noise into channels orthogonal to the intended receiver's channel. Gollakota et al. leveraged cooperative jamming to prevent unauthorized devices from eavesdropping implantable medical devices [20] or key exchange process [18]. Since the evaluation of these schemes focused on the single antenna eavesdropper due to technology constraints, the rapid advancement of MIMO readily destroys the security of original approaches by increasing the number of advisories' antennas. For example, Schulz et al. [40] showed that as long as the eavesdropper has at least as many antennas as the transmitter, the artificial noise can be filtered out via training an adaptive filter with known symbols (e.g. the common protocol headers in WiFi frame). Similarly, the schemes proposed in [20] and [18] were also proved to be vulnerable under multi-antenna attackers.

We note that the existing PHY-layer security approaches mostly aim at preventing eavesdropping against a passive adversary and do not consider active attacks. In the following, we discuss the message integrity protection and authentication that focus on defending against active attacks, and consider to what extent the existing works are able to guarantee these services.

5.2.2 Message Integrity Protection and Authentication

Except for confidentiality, message integrity protection and authentication are two fundamental security services in the wireless communication. Some existing research proposed using out-of-band (OOB) secure auxiliary channels to build message authentication protocols without pre-shared keys [6, 9, 35]. However, an OOB channel would require special hardware and non-trivial human interaction, while its security has been revisited [38]. In addition, whenever keys are stolen or compromised, re-keying involves significant human efforts as well.

Ideally, we want to provide message integrity protection and authentication without relying on pre-shared keys or secure channels. Namely that to establish the veracity of a message and its source using only wireless in-band transmissions. One approach is to use non-cryptographic authentication. Existing approaches on non-cryptographic device authentication can be classified into three categories: (1) Ensuring close device proximity [51], which exploits the channel difference among multiple antennas when devices are in close proximity. However, such techniques require advanced hardware which is not available on all wireless devices. (2) Location distinction, such as temporal link signatures that detect location differences [26, 44]. These techniques require high bandwidth (> 40MHz), which is not always available to low-cost, resource-constrained devices. (3) Device identification [8, 15] which distinguishes devices based on their unique PHY-layer or hardware features. Unfortunately, both location distinction and device identification techniques require prior training or frequent retraining, which is not applicable to networks deployed in uncontrolled environments.

Although the above approaches authenticate a device's presence, they do not necessarily protect the integrity of the message transmitted by a device. There have been a few attempts to achieve in-band message integrity protection [10, 19, 22]. The common underlying idea is to combine ON/OFF keying with unidirectional error detection code. By using this coding method, bit 1 is encoded into ON_OFF slots and bit 0 is encoded into OFF_ON slots. To provide message integrity protection, a data packet is sent first using normal modulation, followed by a cryptographic hash calculated over the message which is encoded using the ON/OFF keying approach (the idea is also shown in Fig. 5.1). The security of this approach is based on the infeasibility of signal cancellation in the wireless channel, which ensures that only unidirectional bit modification is feasible, i.e. attacker could only change OFF slot into ON slot but not in the opposite direction. Besides, according

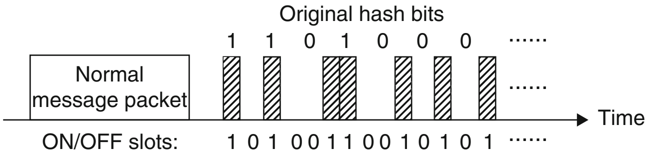


Fig. 5.1 The messaging structure of message integrity protection and authentication

to the second preimage resistance property of Hash functions, it is computationally infeasible for the attacker to compute a new message with the same Hash result. Therefore any tampering with the original message will be detected (w.h.p.). The authentication property is derived based on *authentication through presence* [10], in which the received message is authorized if and only if the receivers verify only one message is received from the intended transmitter and has not been modified. Anti-signal-cancellation is achieved by setting the signal to be random in each ON slot, and based on the assumption that attacker could not extract any knowledge of the source signal and the channel thus it cannot cancel the signal. However, the validity of the infeasibility of signal cancellation assumption was not thoroughly investigated, and in fact, signal cancellation is indeed feasible in many application scenarios [23, 39].

5.2.3 Channel Randomization Based Approaches

Recent studies in wireless communications showed that due to the inherent randomness of the wireless channel such as multipath, even small motions of the antenna can create large variations to the channel [1], which makes channel randomization a promising technology in creating channel advantage. In [21], Hassanieh et al. proposed to randomize the channel via rotating multiple antennas at the transmitter with a fan motor. However, since the fan motor rotates at a constant speed and the number of antennas equipped at the transmitter is relatively small, it is not difficult for the attacker to predict the positions of antennas, and furthermore get sufficient statistical information about the wireless channel to break the scheme. In contrast, randomizing wireless channel via reconfigurable antennas (RAs) are more effective due to the swift reconfigurability and state diversity properties of RAs. [5, 36] adopted RAs to introduce rapid and non-trivial fluctuations to the wireless channel for secret key generation. Their key generation schemes were still based on common randomness with RSS, but the fluctuations of the channel were largely increased by introducing RAs, which led to sufficient independence in RSS profile and high generation rate. However, their methods rely on conventional reconciliation for correcting bit-errors, and thus require an authenticated channel.

Besides defending against passive eavesdroppers, channel randomization can also be used to counter active attacks such as signal manipulation. In our previous work [23], the wireless channel was randomized by using an electric fan blowing the aluminum foil strips attached on the transmitter. Though the message integrity over signal cancellation was protected, the disturbance introduced by a fan is too tiny to defend against strong signal cancellation attack. Typically, when two channels which are close to each other are highly correlated [25, 29], the attacker can cancel out most part of the received signal power via powerful devices with high probability. Besides, randomizing wireless channel via fan is not a systematic approach in practice.

Though channel randomization method has shown to be helpful to enhance PHY-layer security, this technique is still in its infancy, a systematic study of their foundations and applicability is needed. In this chapter, we examine the principles of channel randomization first, followed by exploring its application to prevent active signal manipulation attacks. Specifically, we use reconfigurable antennas to systematically randomize the channel such that it is unpredictable to the active attacker.

5.3 Defending Against Signal Manipulation via Channel Randomization: Theoretical Foundations

5.3.1 Background and Overview

Signal manipulation is a category of active attacks that include symbol injection, flipping, overshadowing, signal cancellation, etc. It directly modifies the message bits at the physical layer such that it is difficult to detect without any high-layer authentication mechanisms. Among them, signal cancellation is the strongest attack which can annihilate the signal all together. Previously, it has been shown by Čapkun et al. [10] that, if signal cancellation over the wireless channel is infeasible, by combining unidirectional error detection codes with ON/OFF keying modulation, we can detect arbitrary signal modifications. However, a practical signal cancellation attack has been demonstrated later by Popper et al. [39], which uses a pair of directional antennas to relay the source signal such that the phase differs by $k\pi$ from the direct signal received by the receiver. It can completely cancel out the received signal, regardless of the message content or modulation. Later Ghose et al. [17] showed the feasibility of signal cancellation not only for one link but also from one to two devices, but infeasible for more than three devices.

Our goal is to provide in-band message integrity protection and authentication, while being resistant against signal cancellation attacks. Note that, signal cancellation is also known as "correlated jamming", since the injected signal is correlated with the legitimate one rather than random. However, due to such difference with traditional jamming, their defenses are completely different. Our goal here is to prevent the legitimate signal from being canceled, while anti-jamming aims at canceling out the external signal. Besides, the key to signal cancellation defense is to increase the signal energy detection probability, while a more powerful traditional jamming signal actually enhances the energy detection probability and helps to protect message integrity. In traditional anti-jamming, the jamming signal can be cancelled out by using digital or mechanical beam-forming and auto-configuration to track the jammer and cancel its jamming signal (for example, Vo et al. [45] and Yan et al. [48]), which will also lead to cancellation of the legitimate signal in our case.

We observe that the key to signal cancellation defense is to prevent an attacker from generating and injecting the correlated signal in the first place. Since the latter is dependent on the physical channel, we need to make attacker's channel different from the legitimate channel. Therefore, proactively and dynamically randomizing the channels can help. This must be done in a principled way such that the legitimate channel is unpredictable to the attacker, while reducing the correlation between the two channels. To this end, we will exploit reconfigurable antennas to change the channel with randomly selected antenna states (radiation patterns), while ensuring de-correlated channels due to heterogeneous antenna gains in different directions.

5.3.2 System Model

In our model, Alice communicates with Bob through a wireless channel. There are two types of transmission modes. In the first one (normal mode) a message is transmitted using standard modulation and data rates, such as 802.11 and OFDM. The second one is called the ON/OFF keying mode, where information bits (like the hash of a normal message) are all encoded using ON/OFF keying combined with unidirectional error detection codes (e.g., Manchester coding). In each ON slot, a normal packet with random content is transmitted, while in OFF slots Alice remains silent. For this mode, Bob uses energy detection to decode the received signal. Periodically (e.g., per symbol interval), Bob obtains a received signal strength (RSS) and compares it with a threshold (α). If the RSS is larger than α for N_s samples then an ON slot is detected. We assume each transmitted signal $x \in \mathbb{C}$ is arbitrary. The channel state information (CSI) $h \in \mathbb{C}$ between Alice and Bob is modeled under Rayleigh fading with additive white Gaussian noise n in outdoor environments, and Rician model in indoor environments.

5.3.3 Threat Model

The attacker's general goal is to break integrity protection, i.e., modify the message without being detected. For the normal mode, we assume the adversary can arbitrarily eavesdrop, inject, modify, replay, and block the message (standard Dolev-Yao model). For the ON/OFF keying mode, we assume an attacker C who knows the exact transmitted signal x, and C's goal is to cancel out the signal received at Bob. To learn x in real-time, C can place a directional antenna closely to the legitimate transmitter A. To create and deliver a correlated signal at B, C will utilize x and her "knowledge" about the CSI h from A to B. Essentially, C possesses a correlated version of h denoted as g (correlation coefficient denoted as $r \in [0, 1]$), as shown in Fig. 5.2.

There are two types of attackers in our model depending on their attack modes. We always assume the attacker cannot replace A or B, or simply block the

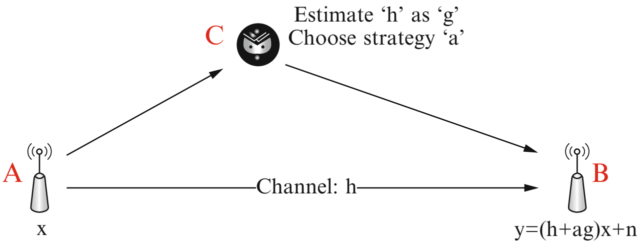


Fig. 5.2 The system model

communication using a Faraday Cage. We do not restrict the number and type of devices the attacker may have. It can either generate its own signals or process and relay the signals from A to B.

Type I This type of attacker can obtain up-to-date and correlated estimation g about A to B's CSI using information from any of the temporal, spatial and frequency domains. For example, it could place multiple receivers close to B, and measure the channel for each transmitted symbol continuously. In the worst case, it obtains the exact A-B channel for every symbol in the past and uses them to predict the future CSI. After estimating h as g, the attacker can decide the cancellation strategy a and send its own signal agx to B.

Type II Note that type I attack model is too theoretical to be used in practice since it requires the attacker to place multiple receivers to measure the channel and combine all estimations, which is costly and computationally complex. Actually, the attacker can easily relay the correlated source signal after processing with one device. Thus, we propose type II attacker to model a more practical attacker. Instead of estimating and predicting future CSI, a type II attacker exploits the intrinsic spatial correlation between channel A-C and A-B, by multiplying the received correlated source signal from A (gx) with cancellation strategy a and relaying it to the receiver via a stable channel (or the other way around). Though in practice, the attacker cannot send its signal to the receiver without any attenuation, the attacker can use powerful directional antennas to relay the processed signal to the receiver, for which the channel can be regarded as stable. Note that the type II attacker is more general than that in [39], since in our model the attacker is capable of processing received signal before relaying it, whereas in [39] the attacker only relays the signal.

It is worth noticing that both types of attack models above are stronger than previous works [10, 19, 22, 39], as the attacker can do real-time signal processing to generate a correlated cancellation signal based on source x and the correlated CSI. In addition, type II is more practical than type I attacker, since it is easier to implement in practice.

5.3.4 *Optimal Strategies for Signal Cancellation Attack and Defense*

5.3.4.1 Game Theoretic Framework

In this section, we theoretically analyze the signal cancellation attack for one symbol in an ON slot. We model the cancellation and anti-cancellation process as a game. The attacker's goal is to transmit a signal correlated with x such that the detection probability P_d of the combined received signal is minimized at B. Therefore we define the attacker's utility function as $U_a = -P_d$. The legitimate pair's strategy is to maximize the energy detection probability and their utility function is $U_l = P_d$.

For the strategy space, let the attacker generate a linear signal [27, 41, 42] that is $agx + v$, in which a is a variable controlled by attacker, g is attacker's knowledge about h (an estimated or correlated version), and v is additive white Gaussian noise with variance σ_v. Thus the overall received signal will be:

$$y = (h + ag)x + n + v \tag{5.1}$$

W.l.o.g., we use the Rician model for A-B channel (Rayleigh model is a special case), note that we choose these models since they are representative and can yield closed-form solutions. In this model, the channel h is composed of two parts: one is the deterministic Line-of-Sight (LoS) component h', the other is the random Gaussian distributed fading component h''. Thus the channel is denoted by $h = h' + h''$.

We assume the attacker could estimate the LoS part precisely. The estimation g is further divided into two parts $g = g' + g''$. The attacker's strategy consists of a tuple $\mathbf{a} = [a', a'', \sigma_v]$ corresponding to each component. Its transmit power can be easily derived based on $\mathbf{a}$, g, the power of x and v, and here we assume it is not bounded.

Under this model, the received signal can be represented by:

$$y = (h' + a'g')x + (h'' + a''g'')x + n + v \tag{5.2}$$

5.3.4.2 Optimal Attack Strategy

Because the LoS and NLoS signal components are independent of each other, the attacker can cancel the two components separately.

A. LoS Component Strategy As the LoS channel component h' is assumed to be precisely known, we have $g' = h'$. Therefore we can easily derive the optimal attack strategy for the LoS component:

Theorem 5.1 *The optimal LoS component cancellation strategy is:*

$$a' = -1 \tag{5.3}$$

The above indicates that the attacker will reverse the LoS signal's phase to completely cancel it out at the receiver side.

B. NLoS Component Strategy Given that the LoS component can be completely canceled, we analyze the optimal attack strategy for NLoS part. We start from deriving the distribution of received power of this component under signal cancellation attack.

According to the type I and II attacker model, the main difference between them is how they are implemented in practice. Thus we can use the same theory to analyze them. In the power expression $P_y = \sigma_x^2(h'' + a''g'')^2 + \sigma_n^2 + \sigma_v^2$, the component $|h'' + a''g''|^2$ follows Gamma distribution $\Gamma(1, 2\sigma^2)$ since $(h'' + a''g'')$ is a CSCG random variable, where $\sigma^2 = \frac{1}{2}E[(h''+a''g'')\overline{(h'' + a''g'')}]$. In addition, the part $\sigma_x^2|h'' + a''g''|^2$ also follows Gamma distribution $\Gamma(1, 2\sigma_x^2\sigma^2)$, because $\sigma_x(h'' + a''g'')$ is a CSCG random variable.

Theorem 5.2 *Given detection threshold α, the probability that a symbol within an ON slot be detected under type I and II attacker's signal cancellation attack is:*

$$P_d(\sigma^2) = e^{-\frac{\alpha-\sigma_n^2-\sigma_v^2}{2\sigma_x^2\sigma^2}} \tag{5.4}$$

According to Eq. (5.4), the detection probability is related to the estimated channel g''. Thus we will first analyze the effect of the parameter σ^2 on the detection probability.

Theorem 5.3 *The NLoS part's optimal signal cancellation attack strategy is:*

$$(a'' = -\frac{E[h''\bar{g}'']}{\sigma_g^2}, \sigma_v^2 = 0) \tag{5.5}$$

The proof is in our previous work [23]. Given the optimal strategy of attacker, we can use Eq. (5.4) in the Appendix of [23] to derive the minimum variance $\sigma_{min}^2 = \frac{1}{2}\sigma_h^2(1 - |r_{h\bar{g}}|^2)$, where $|r_{h''\overline{g''}}|$ is the correlation coefficient. Substitute it into Eq. (5.4), we get the minimum detection probability:

$$P_d(\sigma_{min}^2) = e^{-\frac{\alpha-\sigma_n^2-\sigma_v^2}{\sigma_x^2\sigma_h^2(1-|r_{h''\overline{g''}}|^2)}} \tag{5.6}$$

From Eq. (5.6), we can see that the minimum detection probability decreases with the increase of attacker's correlation coefficient $|r_{h''\overline{g''}}|$. Previous works that either assumed a 0 or 1 correlation coefficient are two extreme cases of our result.

5.3.4.3 Optimal Defender Strategy

Next, we analyze the legitimate pair's optimal strategy. In our model, the signal x is independent of h''. The only transmitter parameter that has an influence on the final detection probability is the power σ_x^2. From Eq. (5.6), we can easily see that the detection probability increases when σ_x^2 increases. In reality, the transmitter's power is limited, thus it indicates that the transmitter should always choose its largest power level to defend against signal cancellation attacks.

5.3.4.4 Simulation Results

We used Matlab to simulate the above theoretical analysis in our previous work [23] and mainly studied the received signal power in the presence of signal cancellation attack. More specifically, in the NLoS Rayleigh fading channels, we generated two CSI sequences with a given correlation coefficient $r_{h\bar{g}}$ to simulate the legitimate channel and attacker's estimation. The transmitting power was $0dB$ and the channel gain was normalized to 1. The signal was modulated using QPSK and the SNR at the receiver side was set to be 25 dB. The attacker was assumed to know $r_{h\bar{g}}$ and σ_g^2 so as to calculate the optimal attack strategy a. Our main simulation results are: (1) The power of received signal achieves the minimum when the attacker applies the proposed optimal attack strategy, which confirms the correctness of our theoretical analysis. (2) There are three factors that could lead to a higher detection probability in optimal cancellation attack: a lower correlation coefficient, a higher detection threshold and a higher transmitting power.

5.4 Channel Randomization Using Reconfigurable Antennas

In this section, we show the crucial criteria in designing our channel randomization approach, of which the basic idea is to randomly switch among RA's different radiation modes every symbol period.

5.4.1 Characteristics of Reconfigurable Antenna

A reconfigurable antenna is an antenna capable of dynamically rearranging its antenna currents or radiating edges in a controlled and reversible manner [7, 12, 50]. For a p-i-n diode based reconfigurable antenna, by changing its structure electronically, it can swiftly reconfigure itself in terms of the radiation pattern, polarization and frequency, or combinations of them. In the aspect of randomizing the wireless channel, we need to prevent the attacker from predicting future CSI from historical CSI sequences (for type I attack), as well as reduce the spatial correlation of CSI (for type II attack). Thus, ideally a RA is expected to have the following two properties for security purposes: (1) the RA should have a large and diverse set of antenna patterns, which provide different gains among different spatial directions (resulting in small spatial correlation); (2) for a given spatial direction, the antenna gains across different antenna modes should have high variations (yielding small temporal correlation).

5.4.2 Antenna Mode Switching Period

For the directional antenna model [4], the CSI is represented as: $h = \sum_{l \in L} f_t(\phi_l, \theta_l) \cdot L_l \cdot f_r(\phi'_l, \theta'_l)$, where L_l is the path gain of the lth path and $f(.)$ is the antenna-specific characterization function which models the transmitter and receiver gain of the direction from which the signal is transmitted and received. Since the antenna gain of a given direction is different under different antenna modes, we can randomize the wireless channel via randomly switching the modes of RA. Moreover, according to a recent study in MIMO [1], the motion of beam steering can change both the LoS and NLoS components of the wireless channel, which also indicates that using RA can create high CSI variations.

Except for increasing the randomness of CSI, to achieve message integrity protection, it is also important to prevent the attacker from predicting future CSI. Consider the scenario that CSI is changing too slowly (that is, one antenna mode lasts for several symbol periods), once obtaining one exact CSI, the attacker is able to cancel out the following symbols that use the same antenna mode. In practice, the attacker is assumed to take at least one symbol period to estimate CSI [2]. To prevent the attacker from accurately predicting future CSI through historical CSI values, the antenna mode of the RA should change at least once in a symbol period. As it is not necessary to change antenna mode too frequently, we let the switching period of antenna mode equal to OFDM symbol duration time in our design.

5.4.3 Multiple Symbols for Message Integrity Protection

We can combine our channel randomization approach with existing message integrity protection schemes as follows. For a general message integrity protection scheme shown in Fig. 5.1, we only need to activate our channel randomization approach during ON slots and synchronization phase, since only those messages need to be protected against cancellation. Considering that the ON slot detection probability grows if there are multiple symbols [22], we can guarantee the energy detection probability of an ON slot by incorporating multiple symbols in it. To do so, we first upper-bound the attacker's knowledge (correlation) under type I and II attack. For the type I attack, the idea is to extract the A-B's CSI by the legitimate receiver B through channel probing, and mimic the attacker's strategy to quantify the intrinsic time-domain correlation in the channel itself, assuming perfect estimation of historical CSI by the attacker. For type II attack, we assume that the attackers can only be located at a certain distance away from the legitimate receiver (and transmitter), which can be implemented by creating a guard zone in practice, otherwise, the attacker can be easily detected. Since the correlation coefficient decreases with the increase of the distance from the attacker to the receiver (this relationship is shown in our previous work [37]), B can estimate the correlation of the channel that is closest to itself (which has most related CSI) to mimic the attacker.

Based on the obtained correlation, we calculate the minimum energy detection probability for each symbol under signal cancellation attack using our theoretical framework. Given a target security requirement (signal cancellation probability for each ON slot is no larger than some threshold), the number of symbols needed in each ON slot can be derived. Then the transmitter applies this parameter during its ON/OFF keying to protect message integrity, while the receiver uses energy detection to recover the source information bits. To enhance efficiency, the transmitter sends a normal message packet followed by Manchester coding and ON/OFF keying of the Hash of the message.

Given the bound of attacker's correlation coefficient, we substitute it along with others parameters (including $\sigma_{h''}$, σ_x, α) into Eq. (5.6). Then we can derive the detection probability P_d for a single symbol, and the minimum necessary number of symbols n in each ON slot:

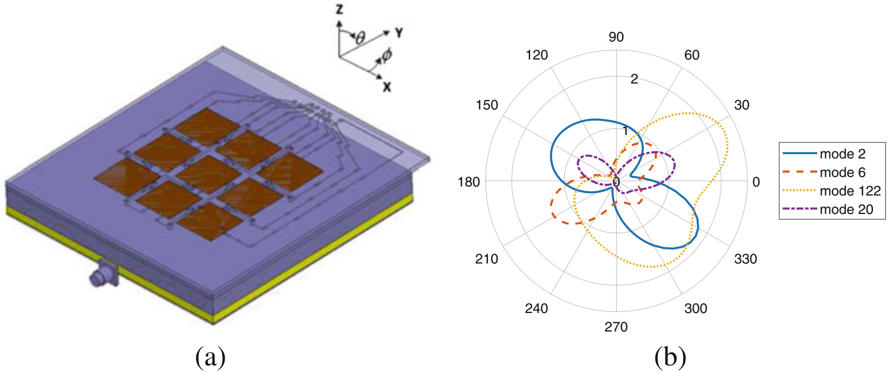
Theorem 5.4 *Given the required minimum detection probability in each ON slot P_s, the minimal number of symbols is:*

$$n = \lfloor log_{1-P_d}^{1-P_s} \rfloor \tag{5.7}$$

5.5 Experimental Results

5.5.1 Channel Randomness and Correlation

To study the impact of attacker's positions on channel correlation when the transmitter is equipped with RA and OA (omnidirectional antenna) respectively, we conduct a preliminary experiment under 246 typical antenna modes that match our reflection coefficient constraint. Figure 5.3a presents the 3D view of RA we use. The reconfigurable parasitic surface consists of 3×3 square-shaped metallic pixels that are connected by 12 p-i-n diode switches [32]. Each switch has ON and OFF status, which brings 4096 possible modes of operation to RA. To show the state diversity of RA, antenna gain in the plane of $\phi = 90°$ for four typical modes is depicted in Fig. 5.3b. We set the distance from the transmitter to the receiver (A-B) to be 120 cm, and change the distance from the attacker to the receiver (C-B). The detailed experiment



(a) (b)

Fig. 5.3 (**a**) 3D view of RA. (**b**) Antenna gain in the plane of $\phi = 90°$

results can be found in our previous paper [37]. The main insights we obtained are: (1) In OA scenarios, wherever the attacker is, the correlation coefficient (≈ 0.98) between A-B and A-C is always quite high, which means the attacker could cancel out most of the transmitted message by just simply relaying its received signal; (2) In contrast, when RA is used, A-B and A-C are much more independent (correlation coefficients are below 0.5 in most cases), which proves that the utilization of RA can increase the randomness between two wireless channels.

To quantify the randomness increment introduced by antenna modes and multipath, we calculate the entropy in terms of antenna gain and CSI. From the results we obtained, we know that: (1) when RA is used, CSI has greater entropy, which corresponds to more randomness of the wireless channel in time-domain; (2) the multipath, noise and other dynamic factors in physical wireless channel lead to the entropy of CSI greater than that of antenna gain (≈ 6.9) and antenna mode ($log_2 246 \approx 7.9$); 3) due to the online nature of signal cancellation attack, to achieve good cancellation performance, the attacker has to estimate the real and imaginary parts of CSI with high accuracy in every symbol period, which is hard to achieve. Thus, even if the CSI distribution has low entropy (e.g., 9 bits), the attacker's average estimation error can still be high.

5.5.2 Attack Effectiveness Evaluation

5.5.2.1 Experiment Setting

We set up three USRP N210 devices with SBX daughter boards using LabVIEW on a table in an indoor lab. We conduct two experiments, where we set the distance between transmitter and receiver (TX-RX) to be 120 cm and 360 cm in experiment 1 (E1) and experiment 2 (E2) respectively. The attacker is put 25.8 cm away from the receiver in both experiments so as to make the attacker get close to the legitimate receiver and meanwhile not be detected. We implement an OFDM transmitter, receiver and attacker, where the transmitter sends packets with known symbols in the 2.45 GHz band with bandwidth set to 100 MHz. All three USRPs are connected with an OctoClock to synchronize their clocks, and eliminate the impact of frequency and phase offset. The QPSK is used and each OFDM symbol contains 320 QPSK symbols. Though our OFDM system has 256 sub-channels, for simplicity, we only estimate the CSI for one of them. The switching time for RA lasts for one OFDM symbol duration, which is 256 μs. To do so, we connect RA with an Arduino Uno Rev 3 programmable microcontroller to randomly switch antenna mode within 4096 available modes.

5.5.2.2 Experimental Strategies

We tested two scenarios for both types of attackers: the transmitter is equipped with OA and RA in scenario 1 and 2 respectively; In both scenarios, the receiver and attacker are equipped with OA. For type I attacker, to generate the attacker's estimated CSI sequence g, we assume the attacker uses a simple autoregression technique to estimate h. That is, the attacker takes the CSI of h at time t_n as the CSI of g at time t_{n+1}. For the much more practical type II attacker, we implement two cancellation attack strategies:

strategy 1: the attacker simply relays the received signal;
strategy 2: the attacker processes the received signal with the proposed optimal attack strategy and then relays it.

5.5.2.3 Evaluation of Cancellation Results

A. Experiment 1 Figure 5.4a, b show the detection probability encountering type II attacker under both strategies in experiment 1. We can see that: (1) In Fig. 5.4b, when the transmitter is equipped with RA, the detection probability after cancellation almost stay the same as before, which shows the effectiveness of channel randomization approach in protecting message integrity; (2) From Fig. 5.4a, we note that the type II attacker who adopts strategy 1 (which is similar to the attacker in [39]) even increases the detection probability in RA scenarios.

For type I attack, we first analyze the channel randomness and correlation. We calculate the auto-correlation coefficient of legitimate CSI sequence and show the result in Fig. 5.5a. We can observe that: (1) the low auto-correlation coefficient of CSI under RA (which is about 0.15) indicates that except for reducing the correlation between two spatial correlated channels, the utilization of RA can also decrease the correlation within CSI sequence in temporal domain; (2) due to the stable indoor environment, the CSI sequence is highly correlated in both temporal and spatial domains when OA is used. Then we implement strategy 2 for type I attacker and show its cancellation performance in Fig. 5.5b. Comparing Fig. 5.5b

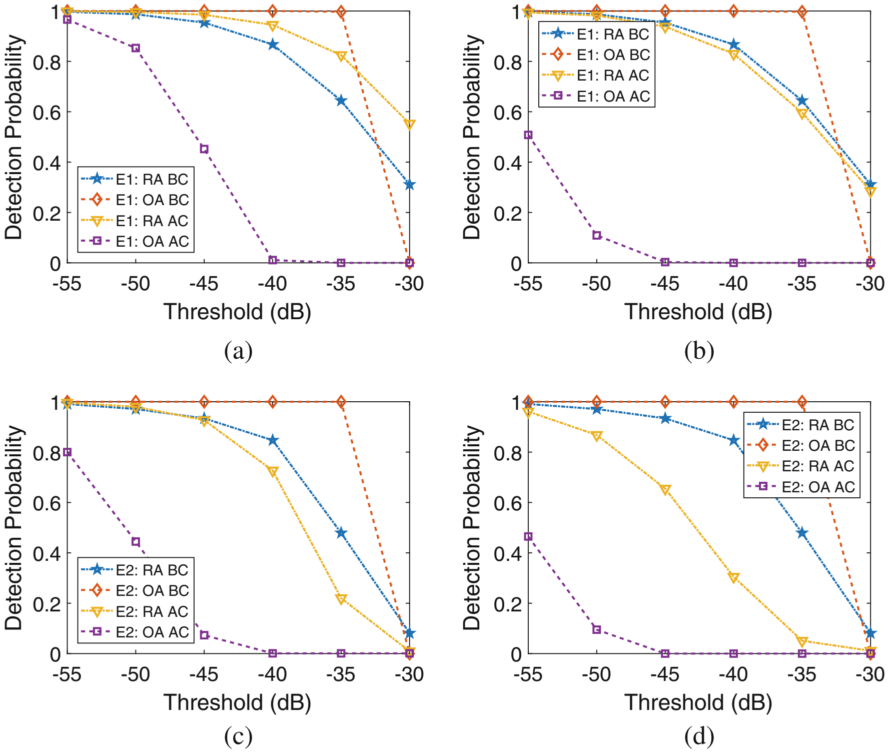


Fig. 5.4 Type II attack, the detection Probability at the receiver. (**a** and **c**) Under strategy 1; (**b** and **d**) under strategy 2 (BC: before cancellation; AC: after cancellation)

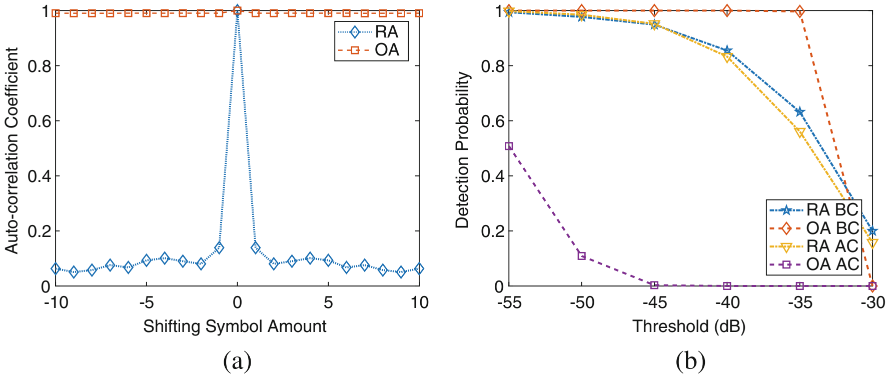


Fig. 5.5 (**a**) Auto-correlation coefficient of legitimate CSI sequence under OA and RA in experiment 1; (**b**) illustrates detection probability encountering type I attacker with strategy 2 in experiment 1

with Fig. 5.4b, we can see that the cancellation performance for type I attacker and type II attacker is similar. However, type II attack is much more practical.

B. Experiment 2 Next, we implement type II attacker for experiment 2, the results are shown in Fig. 5.4c, d. Comparing (a) with (c) and (b) with (d), we can see that the cancellation results for OA are similar. However, when RA is used, the attacker performs better in experiment 2, which indicates the limitation of RA on randomizing wireless channel. Note that the distance of Attacker-RX is the same for both experiments, thus the angle between A-C and A-B in experiment 2 is much smaller than that in experiment 1 due to the increase of the distance between TX and RX. In this case, the antenna gains in the direction of RX and attacker are almost the same, which means the attacker can obtain a highly correlated CSI sequence. Hence we can conclude that when the distance between TX and RX increases, the guard zone at the receiver should increase proportionally to guarantee the effectiveness of the channel randomization approach.

5.5.3 *System Performance*

Considering that the CSI value under some antenna modes of RA can be low, to ensure normal communications after adopting RA, in this part we use the data of experiment 1 to analyze the performance of the message integrity scheme we mentioned in Sect. 5.4.3. More specifically, we first calculate the number of symbols needed in an ON slot from Theorem 5.4. Then we calculate the bit error rate (BER) and link throughput of legitimate pairs under normal communication scenarios with RA and OA respectively. Before presenting the results, we first show the definition of BER and the calculation of link throughput.

5.5.3.1 BER

To clarify, the BER we mentioned here is referred as the error that receiver cannot decode the message (that is, the ON slot in the message is canceled to the OFF slot), changing OFF to ON does not happen because the noise is very small in our experiments. So only OFF_OFF slots are undecodable, which is an error.

5.5.3.2 Link Throughput

If we only consider using the ON/OFF keying mode to carry data, given the number of symbols n, the security requirement P_s and the BER p, we can derive the maximum link throughput between A and B: $c = \frac{1-p}{2\lfloor log_{1-P_d}^{1-P_s} \rfloor \cdot \Delta t}$. If we consider both normal mode and the hash ON/OFF encoding, the maximum throughput will be

$c' = \frac{(1-p)\cdot L_{data}}{T_{data}+2L\cdot\lfloor log_{1-P_d}^{1-P_s}\rfloor\cdot\Delta t}$, where L_{data} and T_{data} are the bit length and transmission time of a normal data packet respectively, while L is hash length. We can see that the higher the per-symbol detection probability P_d, the lower the BER and the higher the throughput.

5.5.3.3 Results

For simplicity, we evaluate the ON/OFF keying mode only. We set the security requirement for successfully detecting each ON slot to be $P_s = 0.9999$. Since the transmitter cannot tell whether there exists the signal cancellation attack or not, to guarantee detection probability, the transmitter always uses the detection probability of a single symbol under optimal attack P_d to calculate the number of symbols needed. Then we calculate the BER and link throughput in normal communications (without cancellation attack).

The results of the number of symbols, BER and link throughput under RA and OA scenarios are shown in Table 5.1. We can have several observations: (1) As the threshold α increases, the energy detection probability in each ON slot decreases, which leads to an increasing number of needed symbols and a decreasing link throughput, but the system is more tolerant to noise/interference; (2) The BER is lower when number of symbols is larger. Note that (1) since the detection threshold is set based on the noise level. The higher the noise level, the higher the threshold should we use, which can decrease the false positive rate for OFF slots. But the tradeoff is that this will decrease the true positive probability (for ON slots) and also the link throughput eventually; (2) the BER for OA scenarios is not exact, because a large number of symbols needed in an ON slot leads to enlarged length of CSI sequences, however, the CSI sequence length in our experiment is 1000, which is not long enough. The value of BER can be remedied by measuring longer CSI sequences in the experiment.

Table 5.1 Number of symbols, BER and link throughput under RA and OA scenarios

	RA			OA		
Threshold (dB)	Number of symbols	BER	Throughput (kbps)	Number of symbols	BER	Throughput (kbps)
– 55	1	0.0060	3.9063	12	0	0.3255
– 50	2	0.0020	1.9531	79	0	0.0494
– 45	3	0	1.3021	3065	–	–

5.6 Conclusion and Future Directions

In this chapter, we explored a proactive and dynamical channel randomization approach to defend against active signal manipulation attacks in the wireless physical layer. We established a signal cancellation attack and defense framework to model the attacker's behavior. Based on the analytical results, we proposed a PHY-layer message integrity protection scheme which uses reconfigurable antennas for channel randomization. Comprehensive experiments were carried out to evaluate the proposed approach under different attack scenarios. Besides defending against signal manipulations, the proposed channel randomization method can also be used to enhance other PHY-layer security objectives, or defend against new attacks. For example, prevent cancellation of the jamming signal by multiple antenna attackers in friendly jamming [43], or known plaintext [40] and ciphertext only attacks [52] against artificial-noise based secret communication schemes, such as orthogonal blinding [3]. For the latter, the key to successful attacks in [40, 52] is the well-trained adaptive filters which filter out the artificial noise. However, the filter is trained over multiple symbol duration, during which the legitimate channel is static. Utilizing our proactive and dynamic channel randomization approach can defeat these attacks by preventing them from successfully training the filter. We will explore both the theoretical foundations and practical schemes to achieve such goals, and fulfill the quest to further understand the applicability and limitations of channel randomization in PHY-layer security.

Acknowledgements This work was partly supported by NSF grants CNS-1410000, CNS-1619728, CAREER Award CNS-1564477, ONR YIP Award N00014-16-1-2650. The multifunctional reconfigurable antenna design aspect of this work performed at Utah State University is supported in part by AFOSR Grant No FA 9550-15-1-0040 DEF.

References

1. F. Adib, S. Kumar, O. Aryan, S. Gollakota, D. Katabi, Interference alignment by motion, in *Proceedings of the 19th Annual International Conference on Mobile Computing & Networking* (ACM, New York, 2013), pp. 279–290
2. S. Ahmadi, *LTE-Advanced: A Practical Systems Approach to Understanding 3GPP LTE Releases 10 and 11 Radio Access Technologies* (Academic, London, 2013)
3. N. Anand, S.J. Lee, E.W. Knightly, Strobe: actively securing wireless communications using zero-forcing beamforming, in *INFOCOM, 2012 Proceedings IEEE* (IEEE, New York, 2012), pp. 720–728
4. E. Anderson, G. Yee, C. Phillips, D. Sicker, D. Grunwald, The impact of directional antenna models on simulation accuracy, in *7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2009. WiOPT 2009* (IEEE, New York, 2009), pp. 1–7
5. T. Aono, K. Higuchi, T. Ohira, B. Komiyama, H. Sasaoka, Wireless secret key generation exploiting reactance-domain scalar response of multipath fading channels. IEEE Trans. Antennas Propag. **53**(11), 3776–3784 (2005)

6. D. Balfanz, D.K. Smetters, P. Stewart, H.C. Wong, Talking to strangers: authentication in ad-hoc wireless networks, in *NDSS* (2002). Citeseer
7. J.T. Bernhard, *Reconfigurable Antennas*. Synthesis Lectures on Antennas, vol. 2(1) (Morgan & Claypool Publishers, San Rafael, 2007)
8. V. Brik, S. Banerjee, M. Gruteser, S. Oh, Wireless device identification with radiometric signatures, in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking* (ACM, New York, 2008), pp. 116–127
9. M. Cagalj, S. Capkun, J.P. Hubaux, Key agreement in peer-to-peer wireless networks. Proc. IEEE **94**(2), 467–478 (2006)
10. S. Čapkun, M. Čagalj, R. Rengaswamy, I. Tsigkogiannis, J.P. Hubaux, M. Srivastava, Integrity codes: message integrity protection and authentication over insecure channels. IEEE Trans. Dependable Secure Comput. **5**(4), 208–223 (2008)
11. V. Casola, A. De Benedictis, M. Albanese, A moving target defense approach for protecting resource-constrained distributed devices, in *2013 IEEE 14th International Conference on Information Reuse and Integration (IRI)* (IEEE, New York, 2013), pp. 22–29
12. B.A. Cetiner, H. Jafarkhani, J.Y. Qian, H.J. Yoo, A. Grau, F. De Flaviis, Multifunctional reconfigurable mems integrated antennas for adaptive mimo systems. IEEE Commun. Mag. **42**(12), 62–70 (2004)
13. I. Csiszár, J. Korner, Broadcast channels with confidential messages. IEEE Trans. Inf. Theory **24**(3), 339–348 (1978)
14. S. Eberz, M. Strohmeier, M. Wilhelm, I. Martinovic, A practical man-in-the-middle attack on signal-based key generation protocols, in *European Symposium on Research in Computer Security* (Springer, New York, 2012), pp. 235–252
15. J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J.V. Randwyk, D. Sicker, Passive data link layer 802.11 wireless device driver fingerprinting, in *USENIX Security Symposium*, vol. 3 (2006), pp. 16–89
16. N. Ghose, L. Lazos, M. Li, Help: Helper-enabled in-band device pairing resistant against signal cancellation. in *26th USENIX Security Symposium*, Vancouver, BC (2017), pp. 433–450
17. N. Ghose, L. Lazos, M. Li, Secure device bootstrapping without secrets resistant to signal manipulation attacks, in 2018 *IEEE Symposium on Security and Privacy (SP)* (IEEE, New York, 2018)
18. S. Gollakota, D. Katabi, Physical layer wireless security made fast and channel independent, in *INFOCOM, 2011 Proceedings IEEE* (IEEE, New York, 2011)
19. S. Gollakota, N. Ahmed, N. Zeldovich, D. Katabi, Secure in-band wireless pairing. in *USENIX Security Symposium*, San Francisco, CA (2011), pp. 1–16
20. S. Gollakota, H. Hassanieh, B. Ransford, D. Katabi, K. Fu, They can hear your heartbeats: non-invasive security for implantable medical devices. in *ACM SIGCOMM Computer Communication Review*, vol. 41 (ACM, New York, 2011), pp. 2–13
21. H. Hassanieh, J. Wang, D. Katabi, T. Kohno, Securing rfids by randomizing the modulation and channel, in *NSDI* (2015), pp. 235–249
22. Y. Hou, M. Li, J.D. Guttman, Chorus: scalable in-band trust establishment for multiple constrained devices over the insecure wireless channel, in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks* (ACM, New York, 2013), pp. 167–178
23. Y. Hou, M. Li, R. Chauhan, R.M. Gerdes, K. Zeng, Message integrity protection over wireless channel by countering signal cancellation: theory and practice, in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security* (ACM, New York, 2015), pp. 261–272
24. S. Jana, S.N. Premnath, M. Clark, S.K. Kasera, N. Patwari, S.V. Krishnamurthy, On the effectiveness of secret key extraction from wireless signal strength in real environments, in *Proceedings of the 15th Annual International Conference on Mobile Computing and Networking* (ACM, New York, 2009), pp. 321–332

25. P.L. Kafle, A. Intarapanich, A.B. Sesay, J. McRory, R.J. Davies, Spatial correlation and capacity measurements for wideband mimo channels in indoor office environment. IEEE Trans. Wirel. Commun. **7**(5), 1560–1571 (2008)
26. A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, A. LaMarca, Ensemble: cooperative proximity-based authentication, in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (ACM, New York, 2010), pp. 331–344
27. A. Kashyap, T. Basar, R. Srikant, Correlated jamming on mimo gaussian fading channels. IEEE Trans. Inf. Theory **50**(9), 2119–2123 (2004)
28. D. Kewley, R. Fink, J. Lowry, M. Dean, Dynamic approaches to thwart adversary intelligence gathering, in *Proceedings of the DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01*, vol. 1 (IEEE, New York, 2001), pp. 176–185
29. P. Kyritsi, D.C. Cox, R.A. Valenzuela, P.W. Wolniansky, Correlation analysis based on mimo channel measurements in an indoor environment. IEEE J. Sel. Areas Commun. **21**(5), 713–720 (2003)
30. L. Lai, Y. Liang, H.V. Poor, A unified framework for key agreement over wireless fading channels. IEEE Trans. Inf. Forensics Secur. **7**(2), 480–490 (2012)
31. S. Leung-Yan-Cheong, M. Hellman, The gaussian wire-tap channel. IEEE Trans. Inf. Theory **24**(4), 451–456 (1978)
32. Z. Li, E. Ahmed, A.M. Eltawil, B.A. Cetiner, A beam-steering reconfigurable antenna for wlan applications. IEEE Trans. Antennas Propag. **63**(1), 24–32 (2015)
33. S. Mathur, W. Trappe, N. Mandayam, C. Ye, A. Reznik, Radio-telepathy: extracting a secret key from an unauthenticated wireless channel, in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking* (ACM, New York, 2008), pp. 128–139
34. U.M. Maurer, Secret key agreement by public discussion from common information. IEEE Trans. Inf. Theory **39**(3), 733–742 (1993)
35. J.M. McCune, A. Perrig, M.K. Reiter, Seeing-is-believing: using camera phones for human-verifiable authentication, in *2005 IEEE symposium on Security and Privacy* (IEEE, New York, 2005), pp. 110–124
36. R. Mehmood, A study of reconfigurable antennas as a solution for efficiency, robustness, and security of wireless systems. Brigham Young University (2015)
37. Y. Pan, Y. Hou, M. Li, R.M. Gerdes, K. Zeng, M.A. Towfiq, B.A. Cetiner, Message integrity protection over wireless channel: countering signal cancellation via channel randomization, in *IEEE Transactions on Dependable and Secure Computing* (2017)
38. T. Perkovic, M. Cagalj, T. Mastelic, N. Saxena, D. Begusic, Secure initialization of multiple constrained wireless devices for an unaided user. IEEE Trans. Mob. Comput. **11**(2), 337–351 (2012)
39. C. Pöpper, N.O. Tippenhauer, B. Danev, S. Capkun, Investigation of signal and message manipulations on the wireless channel, in *European Symposium on Research in Computer Security* (Springer, New York, 2011), pp. 40–59
40. M. Schulz, A. Loch, M. Hollick, Practical known-plaintext attacks against physical layer security in wireless mimo systems, in *NDSS* (2014)
41. S. Shafiee, S. Ulukus, Capacity of multiple access channels with correlated jamming, in *Military Communications Conference, 2005, MILCOM 2005* (IEEE, New York, 2005), pp. 218–224
42. S. Shafiee, S. Ulukus, Mutual information games in multiuser channels with correlated jamming. IEEE Trans. Inf. Theory **55**(10), 4598–4607 (2009)
43. N.O. Tippenhauer, L. Malisa, A. Ranganathan, S. Capkun, On limitations of friendly jamming for confidentiality, in *2013 IEEE Symposium on Security and Privacy (SP)* (IEEE, New York, 2013), pp. 160–173
44. A. Varshavsky, A. Scannell, A. LaMarca, E. De Lara, Amigo: proximity-based authentication of mobile devices, in *International Conference on Ubiquitous Computing* (Springer, New York, 2007), pp. 253–270

45. T.D. Vo-Huu, E.O. Blass, G. Noubir, Counter-jamming using mixed mechanical and software interference cancellation, in *Proceedings of the Sixth ACM Conference on Security and Privacy in Wireless and Mobile Networks* (ACM, New York, 2013), pp. 31–42
46. Q. Wang, H. Su, K. Ren, K. Kim, Fast and scalable secret key generation exploiting channel phase randomness in wireless networks, in *INFOCOM, 2011 Proceedings IEEE* (IEEE, New York, 2011), pp. 1422–1430
47. A.D. Wyner, The wire-tap channel. Bell Labs Tech. J. **54**(8), 1355–1387 (1975)
48. Q. Yan, H. Zeng, T. Jiang, M. Li, W. Lou, Y.T. Hou, Jamming resilient communication using mimo interference cancellation. IEEE Trans. Inf. Forensics Secur. **11**(7), 1486–1499 (2016)
49. C. Ye, S. Mathur, A. Reznik, Y. Shah, W. Trappe, N.B. Mandayam, Information-theoretically secret key generation for fading wireless channels. IEEE Trans. Inf. Forensics Secur. **5**(2), 240–254 (2010)
50. X. Yuan, Z. Li, D. Rodrigo, H.S. Mopidevi, O. Kaynar, L. Jofre, B.A. Cetiner, A parasitic layer-based reconfigurable antenna design by multi-objective optimization. IEEE Trans. Antennas Propag. **60**(6), 2690–2701 (2012)
51. K. Zeng, K. Govindan, P. Mohapatra, Non-cryptographic authentication and identification in wireless networks [security and privacy in emerging wireless networks]. IEEE Wirel. Commun. **17**(5), 56–62 (2010)
52. Y. Zheng, M. Schulz, W. Lou, Y.T. Hou, M. Hollick, Profiling the strength of physical-layer security: a study in orthogonal blinding, in *Proceedings of the 9th ACM Conference on Security & Privacy in Wireless and Mobile Networks* (ACM, New York, 2016), pp. 21–30

Chapter 6
Towards High-Resolution Multi-Stage Security Games

Aron Laszka, Xenofon Koutsoukos, and Yevgeniy Vorobeychik

Abstract In recent years, we have seen a large number of cyber-incidents, which demonstrated how difficult it is to prevent cyber-breaches when facing determined and sophisticated attackers. In light of this, it is clear that defenders need to look beyond the first lines of defense and invest not only into prevention, but also into limiting the impact of cyber-breaches. Thus, an effective cyber-defense must combine *proactive defense*, which aims to block anticipated attacks, with *reactive defense*, which responds to and mitigates perceived attacks (e.g., isolating and shutting down compromised components). However, planning defensive actions in anticipation of and in response to strategic attacks is a challenging problem. Prior work has introduced a number of game-theoretic security models for planning defensive actions, such as Stackelberg security games, but these models do not address the overarching problem of proactive and reactive defenses in sufficient detail. To bridge this gap, we introduce a modeling approach for building high-resolution multi-stage security games. We describe several approaches for modeling proactive and reactive defenses, consider key modeling choices and challenges, and discuss finding optimal defense policies. With our study, we aim to lay conceptual foundations for developing realistic models of cyber-security that researchers and practitioners can use for effective cyber-defense.

A. Laszka
Department of Computer Science, University of Houston, Houston, TX, USA
e-mail: alaszka@houston.edu

X. Koutsoukos
Department of Electrical Engineering and Computer Science, Vanderbilt University, Nashville, TN, USA
e-mail: xenofon.koutsoukos@vanderbilt.edu

Y. Vorobeychik (✉)
Computer Science and Engineering, Washington University, St. Louis, MO, USA
e-mail: yvorobeychik@wustl.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances
in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_6

6.1 Introduction

Traditionally, security research has focused on preventing attackers from breaching the security of a system or network. While researchers are making considerable advances in this direction, attack techniques are also evolving, which makes providing security an uphill battle. Indeed, attaining perfect security remains virtually impossible for practical systems. The number of reported cyber-incidents is increasing steadily, and the total cost of malicious cyber-activities to the U.S. economy has recently been estimated to be between $57 and $109 billion [9]. For instance, according to a 2017 industry report, 67% of companies with critical infrastructure suffered at least one attack in the preceding 12 months; in particular, 91% of power companies have experienced an attack [17].

In light of this, it is clear that defenders cannot focus only on the first lines of defense, and they must look beyond the prevention of cyber-breaches. Besides preventing breaches, defenders can also alleviate cyber-security risks by reducing the expected impact of successful attacks. In practice, there are a number of actions that defenders may take to limit the impact of a breach, such as quickly isolating and shutting down compromised hosts or reconfiguring the uncompromised ones. While these actions cannot prevent an attack, they can mitigate it before it could cause significant damage. We will refer to such actions collectively as *reactive defense* approaches to emphasize that these actions are taken in response to perceived (or suspected) cyber-attacks.[1]

An effective cyber-defense must combine this reactive approach with *proactive defense* actions. Proactive defense includes actions taken in anticipation of an attack, such as finding and patching software vulnerabilities before an adversary could exploit them. Optimal cyber-defense must consider the whole spectrum of available proactive and reactive actions, and it must implement them in a combination that minimizes cyber-security risks. However, real defenders typically have a finite budget, which limits the amount of resources, effort, and time available to them for implementing cyber-defenses. Consequently, they need to carefully plan what proactive actions to implement in anticipation of attacks and what reactive actions to implement under various attack scenarios in order to minimize cyber-risks subject to their budget constraints.

A key factor in this planning problem is the strategic nature of cyber-security. The most threatening, sophisticated attacks are very often strategic in the sense that adversaries tailor their malicious actions to the defenders' plans. In light of this, defenses must also be planned strategically: On the one hand, defenders must anticipate attacks and plan their actions accordingly, assuming that adversaries will adapt. On the other hand, defenders have to react to observed attacks to mitigate them (e.g., isolate and re-install compromised hosts), assuming that adversaries have mounted strategic attacks and are ready for strategic escalation.

[1] Note that we use the term "reactive defense" to refer to actions taken in response to perceived or suspected attacks. This is different from planning defenses in response to risks, which is sometimes referred to using similar terms (e.g., responsive or reactive security).

Such strategic interactions between defenders and attackers are modeled most naturally using game theory. Indeed, a number of game-theoretic models have been proposed for studying the defense of networked systems [32, 38]. However, prior work has not addressed the overarching problem of proactive and reactive defenses in sufficient detail. Firstly, a number of research efforts have studied high-level models of cyber-security, but these papers often consider very abstract notions of security investments (e.g., allocation of abstract defensive resources to targets [30]). Further, these models are typically based on two-stage security games, which consider only proactive actions, but not reactive ones. Secondly, a number of research efforts have studied the optimal implementation of particular actions in detail, some even considering continuous conflicts and reactive approaches (e.g., resetting potentially compromised computational resources [51]). However, these models typically include only one particular type of action, and it is often unclear—especially for practitioners—how to combine different types of actions most effectively.

To bridge this gap, we discuss how to build realistic, high-resolution multi-stage security games for networked systems, which can form a conceptual foundation for the optimal implementation of proactive and reactive defenses. We first consider the most widely used class of security models, called Stackelberg security games, and argue that these are not well suited for studying reactive defenses. We then discuss stochastic games, which provide a general mathematical framework for modeling multi-stage interactions. Based on this framework, we introduce our approach for modeling the proactive and reactive defense of networked systems against strategic attacks, focusing on key modeling choices and challenges. Then, we describe canonical types of proactive (redundancy, diversity, isolation, hardening, and detection) and reactive (islanding, resetting, and reconfiguration) defenses, again focusing on key modeling choices and challenges. Finally, we consider the problem of finding optimal defense strategies in our model, which is generally a computationally hard problem, and discuss reinforcement learning as a promising solution approach.

The remainder of this chapter is organized as follows. In Sect. 6.2, we describe two-stage Stackelberg game model of security. In Sect. 6.3, we discuss stochastic games for modeling multi-stage interactions in security. In Sect. 6.4, we introduce our modeling approach for building realistic multi-stage models of security. In Sects. 6.5 and 6.6, we describe canonical approaches for proactive and reactive defenses, respectively, and we discuss how to model them. In Sect. 6.7, we consider how to solve realistic multi-stage security games and find optimal defense strategies. In Sect. 6.8, we provide concluding remarks.

6.2 Stackelberg Game Models of Security

A very natural game theoretic model of security, which has received considerable attention in recent years, is known as *Stackelberg games* [30, 50]. A Stackelberg game involves two stages: in the first stage, the defender chooses a defensive posture (such as which vulnerabilities to patch, or how to configure the firewall), and in the second stage, the attacker chooses the best attack. The crucial feature of this model is that the attacker is assumed to observe the defensive decision; while in reality this is a very strong assumption, it is also a sound starting point for analysis, as it makes a worst-case assumption about the information available to the attacker, in the spirit of Kerckhoffs's principle [28].

Formally, let D and A denote the sets of actions available to the defender and attacker, respectively, with $d \in D$ and $a \in A$ referring to a particular defense/attack action. To allow for the possibility that the defender randomizes, we let S be the *strategy* set of the defender. Thus, the defender may *commit* to an action, in which case $S = D$, or may be able to commit to a probability distribution over D, in which case $S = \Delta(D)$, where $\Delta(D)$ is the set of all probability distributions over D. Next, we define the utility functions $u_D(s, a)$ and $u_A(s, a)$ for the defender and attacker, respectively, where $s \in S$ is the defender's strategy.

Suppose that the defender commits to a strategy $s \in S$. The attacker's *best response* to s is $\phi(s) \in \text{argmax}_{a \in A} u_A(s, a)$. Correspondingly, the defender then aims to find a strategy s which maximizes its payoff *given* the attacker's best response function $\phi(s)$. In particular, a pair of defender and attacker strategies (s^*, $\phi^*(s)$) is a *Stackelberg equilibrium* if $\phi^*(s)$ is an attacker's best response for each s, and $s^* \in \text{argmax}_{s \in S} u_D(s, \phi^*(s))$. This equilibrium concept raises a subtle but important issue of tie-breaking for the attacker. A common way to resolve it is to consider a *Strong Stackelberg equilibrium (SSE)* in which the attacker breaks ties in the defender's favor [50].

Stackelberg game models of security that we described above are clearly simplistic: in these models, the world has only two stages, with the defender making the first decision, followed by the attacker. In real security settings, the game involves many such stages. For example, after the attacker chooses an attack, once the attack has been observed, the defender can deploy additional mitigations, such as updating anti-virus software, patching vulnerabilities which have been exploited, and rebuilding the compromised machines. The attacker, in turn, can subsequently react to such measures, for example, by exploiting another vulnerability, and so on. A common and very general framework for capturing such multi-stage interactions is through the formalism of *stochastic games*, which we describe next. However, as we subsequently point out, stochastic games are *too* general, and fail to capture much structure exhibited in realistic problems. Consequently, we suggest moving to less general, high-resolution models of multi-stage interactions, which allow us to make progress towards applying game theoretic tools to realistic security scenarios.

6.3 Stochastic Games in Security

A stochastic game is a very general mathematical framework for modeling multi-stage interactions. In the context of security, a two-player stochastic game has a finite set of states X, finite sets of actions for the defender D and attacker A, a transition function $P_{xx'}^{da} = \Pr\{x'|x, d, a\}$, and immediate reward functions $u_D(d, a; x)$ and $u_A(d, a; x)$ for the defender and attacker, respectively [16, 52].

The game proceeds in discrete time steps $t = \{0, 1, 2, \ldots\}$, where the state at time t, denoted by x_t, is determined stochastically according to the transition function, given the previous state x_{t-1} as well as the previous actions d_{t-1} and a_{t-1} that were taken by the players. The state x_t along with the actions d_t and a_t taken in that state then determine the players' immediate rewards. Let T be the time horizon of the game; it is finite if the game has a finite horizon, and infinite otherwise. Let the history of states and player actions through time T be $h = \{x_0, d_0, a_0, \ldots, x_T, d_T, a_T\}$ Then, we define the realized utility of a player $i \in \{D, A\}$ (attacker or defender) to be

$$\tilde{U}_i(h) = \sum_{t=0}^{T} \gamma^t \cdot u_i(d_t, a_t; x_t),$$

where $\gamma \in [0, 1]$ is the discount factor, which weighs distance rewards exponentially less than current.[2] Since history is stochastic, we can define *expected* utility of player $i \in \{D, A\}$ starting in state x as

$$U_i(x) = \mathbb{E}_h[\tilde{U}_i(h) \mid x_0 = x].$$

An important and well-known result in (discounted) stochastic games is that there always exists an equilibrium in which player strategies depend only on current state and, in finite horizon games, time. Specifically, let a policy π_i of a player i determine the action this player takes at each time step of the game. In an infinite-horizon stochastic games, there is an equilibrium pair of policies (π_D, π_A) such that π_i depends only on state x; in finite-horizon stochastic games, such policies would also depend on the time step t.

There are two variations of stochastic games which are particularly relevant to multi-stage interactions in security. One, which is a special case of the stochastic game formalism above, involves alternating moves by the defender and attacker, in which the defender moves first. The significance of this model is that it is a natural extension of the standard two-stage Stackelberg game: indeed, a Stackelberg game model is just such a game with a horizon $T = 1$ (so that there are only two time steps, 0 and 1). Clearly, this extension captures in a very general way the intuition that we started with: the game between a defender and an attacker extends beyond two steps, with a defender reacting to an observed attack, the attacker subsequently reacting to the defender, and so on. A simple way to encode such an iterative encounter in the stochastic game formalism is as follows. Let state x encode which player's turn it is to move; we can do this by adding a binary state variable $x_m \in \{0, 1\}$, which deterministically flips in each step. We can let $x_m = 0$ when it's the defender's turn to move, and $x_m = 1$ when the attacker moves. Additionally, let us extend the action sets of both players to allow them to depend on state. Thus, the defender's

[2] We chose the discounted version of the stochastic game here as we view it as the best model of security interactions, where players are sensitive to time. For example, other things being equal, an attacker would rather obtain intellectual property data sooner than later.

set of actions is $D(x)$ and the attacker's set is $A(x)$; this change has no effect on the theoretical properties of equilibria in stochastic games that we had noted above. Thus, whenever $x_m = 0$, $A(x) = \emptyset$ and, conversely, when $x_m = 1$, $D(x) = \emptyset$.

Another variation, which is actually (and somewhat surprisingly) qualitatively different from the conventional stochastic games, is the notion of *Stochastic Stackelberg games (SSGs)* [52, 53]. The definition of SSGs is nearly identical to stochastic games, with one crucial difference: first, the defender commits to a policy π_d, and then the attacker best responds with its own policy $\pi_a(\pi_d)$—note that now the attacker's policy can be different depending on which policy the defender commits to! It turns out that this difference makes SSGs dramatically more challenging to analyze and solve, in general [52, 53]. For example, it is no longer the case that we can restrict attention to policies for both players which only depend on current state, *even when the horizon is infinite* [52].

At this point, we have described several very general formalisms which allow us to capture multi-stage interactions in security. The major concern with these, however, is that they are *too* general: indeed, stochastic games are difficult to solve even when the state space is relatively small. In practice, the number of variables which determine state can be substantial, and even the representation of a stochastic game described above becomes intractable. Clearly, in order for us to significantly advance the art in considering interesting multi-stage interactions, we need to consider lower-level structure. In the remainder of this chapter, we propose and illustrate the idea of *high-fidelity* multi-stage games, that is, games in which we make use of much more specialized, high-fidelity models of the domain. While this necessarily loses generality, we argue that such modeling is necessary to reveal important structure in multi-stage games which can enable us to solve more realistic problems.

6.4 Towards Realistic Multi-Stage Game Models

We now discuss modeling approaches and assumptions for high-fidelity multi-stage games for studying the defense of networked systems. While our discussion will consider networked systems in general, we will use *cyber-physical systems* (CPS) as a running example to illustrate the practical applicability of our model. Defending CPS from cyber-physical attacks is an issue that is both pressing and challenging. As CPS are becoming more prevalent (e.g., smart grid, Industrial Internet of Things), the importance of ensuring that they are resilient to cyber-attacks is growing rapidly. For example, cyber-attacks against critical infrastructures, such as smart water-distribution and transportation networks, pose a serious threat to public health and safety. Indeed, real-world attacks have demonstrated that cyber-attacks may penetrate CPS and cause significant physical damage [2, 37, 44, 49, 56]. On the other hand, defending a complex and large-scale CPS, such as smart critical infrastructure, is extremely challenging. These systems often face a variety of threats, contain low-power and legacy components, have large attack surfaces, and have a number of undiscovered software vulnerabilities in their sizable codebases. In light of this, defending CPS is an ideal application example for security game models.

6.4.1 System Model

We first introduce a basic model of networked systems, which will provide a basis for the discussion of game-theoretic models. In general, we can model a networked system as a graph (C, E), where C is a set of components and E is a set of links between the components. Depending on the granularity of the model, a component may correspond to a subnetwork, a host, a running process, or just a software module. A link models a communication channel between two components, which can be either physical (e.g., wired link) or logical (e.g., VPN). A link may be either unidirectional or bidirectional, which we can model using either directed arcs (i.e., $E \subseteq C \times C$) or undirected edges (i.e., $E \subseteq \binom{C}{2}$). A key factor in network security is that links are not only used to transmit information, control signals, etc., but they may also be exploited by an attacker to escalate an attack by compromising the neighbors of an already compromised component.

CPS Example To illustrate how we can model a networked system as a graph, we now present a high-level model of networked cyber-physical systems, focusing on the cyber parts of the systems. Figure 6.1 shows an example of a networked cyber-physical system, consisting of a variety of physical devices. We let the components C model such physical devices, which we divide into four component types:

- *sensor*: components that measure the state of physical processes (e.g., water-pressure sensors, induction-loop sensors for measuring traffic);
- *actuator*: components that directly affect physical processes (e.g., valves, pumps, circuit breakers);

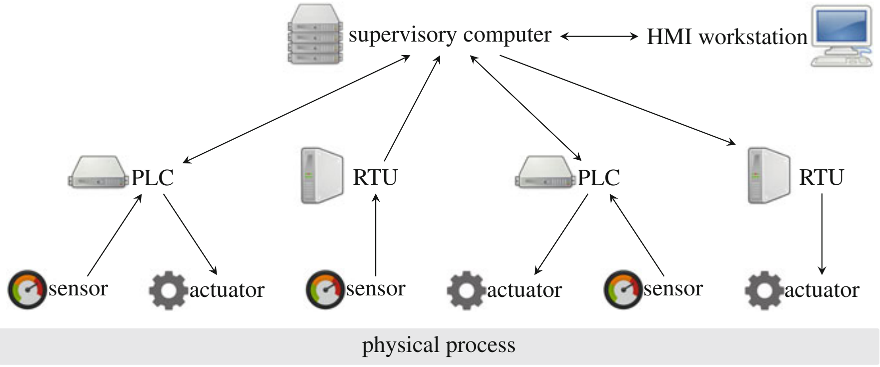


Fig. 6.1 Example cyber-physical system. Labeled icons represent components; arrows represent links through which sensor data and control signals can flow

- *processing*: components that process and store data and control signals (e.g., PLCs, RTUs, supervisory computers);
- *interface*: components that interact with human users (e.g., HMI workstations) or other systems, which are not part of the model.

The links E model communication links between these devices, which are used to transmit sensor data and control signals. The observability and controllability of the physical processes within the CPS depend not only on the functionality of the individual components, but also on the structure of the graph (C, E). Depending on this structure, an attacker may be able cause physical damage or loss by compromising a subset of the components, and tampering with sensor data or control signals.

6.4.2 Game-Theoretic Model

We next discuss how to build a stochastic security game based on the above model of networked systems. We first consider the players' action sets D and A, and then their utility functions u_i.

We assume alternating moves by the defender and the attacker, the defender moving first. In each time step, a player may take multiple actions. Slightly abusing notation, we let D and A denote the sets of actions available to the defender and attacker, respectively. A policy π_i determines the subset of actions to be taken in each time step.

We divide the defenders' actions D into two disjoint sets of actions (see Table 6.1):

- *Proactive defense actions*D_P: Proactive actions are taken in anticipation of attacks (e.g., deploying an intrusion detection system). In our model, we assume that the defender can take these actions only in time step $t = 0$, which represents everything that happens before the attacker may mount its attack. We discuss proactive actions in more detail in Sect. 6.5.
- *Reactive defense actions*D_R: Reactive actions are taken in response to an observed attack (e.g., shutting down and re-installing a compromised host). In our model, we assume that the defender can take these actions only in time steps $t > 0$. We discuss reactive actions in more detail in Sect. 6.6.

Meanwhile, an attacker tries to compromise or impair the components of the system by attacking them.[3] We let $C_t^C \subseteq C$ and $C_t^I \subseteq C$ denote the sets of components that are compromised or impaired by the attacker at the end of time step t. Each attack—of which the attacker may mount multiple in a time step—targets a specific subset $K \subseteq C$ of components using a specific attack method (e.g.,

[3] For ease of presentation, we only consider attacks against components, but it would be straightforward to extend our modeling approach to also consider attacks against links.

Table 6.1 Defense actions

Type	Name	Idea	Section
Proactive	Redundancy	Deploying redundant components	6.5.1
	Diversity	Implementing components using a diverse set of hardware and software	6.5.2
	Isolation	Removing links between components	6.5.3
	Hardening	Making components (or implementation types) more resilient to attacks	6.5.4
	Detection	Deploying intrusion detection systems	6.5.5
Reactive	Islanding	Removing links between components	6.6.1
	Resetting	Resetting components into known secure states	6.6.2
	Reconfiguration	Changing the configuration of components	6.6.3

code injection attack or DDoS attack). The set of attack actions A corresponds to the possible combinations of targeted components and attack methods. Attacks are non-deterministic in the sense that they do not necessarily succeed in compromising or impairing all the targeted components K. For example, an attack might require finding a software vulnerability in a certain implementation or guessing a password, and the attacker might fail to do so. In general, the success probability of an attack is an increasing function of the set of components that have already been compromised or impaired.[4] Firstly, the attacker might exploit the implicit trust relations between components that are connected by links E to easily compromise the neighbors of an already compromised component. Secondly, the impairment of components may result in cascading failures, which makes the impairment of other components easier.

The attacker also incurs a cost for mounting its attacks. The cost of mounting an attack depends on both the set of targeted components K and the method of attack. For attack methods that are easily replicated for a large number of components (e.g., once a software vulnerability is found, the attacker may easily compromise a large number of hosts), the cost can be modeled as a submodular function of K, capturing the diminishing marginal cost of attacking an additional component.

In general, the attacker's goal is to cause damage or gain some benefit by compromising or impairing the components of the system, while the defender's goal is to minimize its losses due to successful cyber-attacks. These goals are captured using the players' utility functions u_i, which they aim to maximize through their action choices. In principle, we can express the defender's utility as the baseline utility provided by an operational system minus the losses caused by the attacks and the costs of implementing defensive actions. Note that since this baseline utility does not depend on the players' actions, it may be omitted without affecting the best-

[4] In practice, the probability may decrease since the defender may notice a large-scale attack and deploy countermeasures in response. In our model, this effect is captured explicitly through the defender's reactive actions.

response or equilibrium strategies. Similarly, we can express the attacker's utility as the attacker's gain from compromising or impairing the components minus the costs of mounting its attacks. The form of the loss and gain functions depends on the specifics of the modeled system. However, in most systems, we can express loss and gain as functions of the compromised and impaired components C_t^C and C_t^I; hence, we may express a player's utility u_i for time step t as a function $u_i(d_t, a_t; C_t^C, C_t^I)$. In a simple model, we may also assume that the defender's loss and the attacker's gain are always equal. Note that even under this assumption, the game is not necessarily zero sum since the players also incur costs for their actions.

Finally, while it is impossible to provide generic loss/gain functions that are applicable to all system, we can provide modeling guidelines. For attacks against confidentiality, loss/gain may be expressed as a submodular function of the set of compromised components C_t^C since the information gained from compromising more and more components exhibits a diminishing return due to the possible overlap between the information contained by a set of components.[5] On the other hand, for attacks against integrity, loss/gain may be expressed as a supermodular function of the set of compromised components C_t^C since when information is stored redundantly on multiple components, the attack remains undetected only if all of these components are compromised. Similarly, system functionality that is provided redundantly by multiple components can be tampered with (or disabled) only by compromising the majority of the components (or impairing all of them).

CPS Example In many cyber-physical systems, loss can be measured in terms of physical impact. For example, a cyber-attack against a smart transportation network may cause disastrous traffic congestion [35, 55].[6] Such attacks have been made possible by the evolution of traffic control from standalone hardware devices into complex networked systems, which has exposed traffic control to attacks through wireless interfaces or even remote attacks through the Internet. As demonstrated by the 2006 incident in Los Angeles, tampering with traffic control can cause significant losses through congestion [23].

To formulate a multi-stage security game for smart transportation networks, we may model the physical part of the system using an established traffic model (e.g., Daganzo's well-known cell-transmission model [10, 11]), while we can model the cyber part of the system using the following components C:

- *interface*: human-machine interface components, which traffic operators can use to control traffic lights in the transportation network;
- *processing*: devices that process and forward traffic control signals;
- *actuator*: traffic lights with software-based controllers.

[5] Defender's may turn this around by using, e.g., secret sharing schemes, which lead to a supermodular loss/gain functions for confidentiality. This possibility is considered explicitly among the defender's proactive actions; here, we consider a baseline case without such schemes.

[6] In practice, due to hardware-based failsafes, compromising a traffic signal does not allow an attacker to set the signal into an unsafe configuration that could immediately lead to traffic accidents [21].

An attacker may try compromise these components, e.g., by connecting to traffic lights through local-area wireless networks and exploiting software vulnerabilities. Indeed, studies have found that many traffic control devices that are deployed in practice have unpatched known software vulnerabilities [21, 55]. Once an attacker has compromised some set of the components C^C, it can alter the schedules of traffic lights, thereby causing disastrous traffic congestion. We can quantify the impact of such an attack as the total increase in travel time experienced by all the drivers. Assuming a malicious attacker who is interested in maximizing the defender's loss, we can measure both the defender's loss and the attacker's gain as the impact of the attack.

6.4.3 Imperfect and Incomplete Information

A key aspect of security games is that generally the players do not possess perfect and complete information. Firstly, the players might not know what actions their opponents have taken and, hence, which components are compromised, which means that they possess *imperfect information*. While imperfect information can affect both players, there is often an asymmetry between the players, which may put the defender at a grave disadvantage. On the one hand, the attacker knows which components it has attacked and—in most cases—which components it has compromised. On the other hand, the defender may not immediately learn of compromises. Indeed, a recent study has found that on average, it takes 191 days to detect a data breach [45]. Lack of perfect information can prevent the defender from reacting and implementing countermeasures in time to mitigate an attack, which enables the attacker to operate covertly in the compromised system, causing damage and extracting information. In practice, attackers often seek to remain covert for as long as possible in order to cause more damage or extract more information over a longer period of time. For example, sophisticated spyware (e.g., used in state-sponsored cyber-espionage campaigns) often remain covert for extended periods of time [26]. Even malware that causes physical damage in a cyber-physical system may remain covert for months, as demonstrated by the Stuxnet worm [27].

To some extent, the attacker might also suffer from imperfect information. While we typically assume—following Kerckhoffs's principle—that the attacker can learn the defender's strategy, this strategy may be a probability distribution over possible actions, and the attacker does not learn the specific action if it chosen truly randomly. Further, the attacker might also not be able to directly observe which components it has compromised. For example, the defender might secure a host (e.g., shutdown and re-install) that is not connected to the Internet, which the attacker has compromised earlier using a worm. In such a scenario, the attacker will not learn immediately that the component is no longer compromised (or if it ever were). In light of this, the players' policies cannot be defined as functions of the state x. Rather, each players' policy needs to be defined as a function of their observations.

In addition to imperfect information, the players may also suffer from incomplete information, i.e., not knowing the exact action sets, state transition functions, or utility functions. Firstly, the defender might not know what actions are available to an attacker (e.g., specific attack techniques and exploits) or how likely these actions are to successfully compromise or impair components. Further, the defender might also not know the attackers' objectives and what resources they have (e.g., script kiddies or nation-state sponsored attackers) [13, 14]. Secondly, the attacker might not have complete knowledge of the target system. Even though we follow Kerckhoffs's principle and assume that the attacker will be able determine the design of the system, the defender might still be able to deceive the attacker [47]. For example, the defender might deploy honeypots in the system in order to waste the attacker's effort and observe its behavior [43].

6.5 Proactive Defense

Proactive defense includes actions taken by a defender in anticipation of attacks. Here, we discuss various approaches for the proactive defense of networked systems in more detail, focusing on how to incorporate them into our game-theoretic model. Recall from our previous discussion that these actions are taken in time step $t = 0$ (i.e., before the attacker's first move).

6.5.1 Redundancy

Redundancy means deploying additional components in the system, which are not necessary for providing required system functionality or performance [5]. When facing denial-of-service attacks, which impair components, the benefits of redundancy are clear: in case of an attack, the redundant components may be used instead of the ones that are unavailable due to the attack. As long as a sufficient set of components are still available, the system might suffer from decreased performance, user experience, etc., but retains its functionality.

In practice, redundancy may be implemented, for example, by deploying additional physical hosts or storing redundant copies of information. In a fine-grained model, where components correspond to software modules or services, redundancy can be implemented even for security mechanisms. For example, multi-factor authentication methods grant a user access to a system only after verifying the user's identity using multiple authentication methods [12]. In a cyber-physical system, redundancy can be implemented by, e.g., deploying multiple sensors for monitoring the same physical process [1], or deploying multiple controllers and letting actuators act based on the median control value provided by these controllers.

While the benefits of redundancy are obvious in the case of denial-of-service attacks, they are much less straightforward in the case of integrity attacks that compromise and tamper with components. Since defenders—and the systems under their control—may not know which components have been compromised, when redundant components provide contradictory information, they face the challenging problem of deciding which components to trust. Further, simple redundancy might even increase risks when it comes to confidentiality. Without redundancy, the attacker would need to compromise a particular component to gain a particular piece of information. However, with redundancy, it needs to compromise one out of many redundant components, which may give the attacker more opportunities to succeed. Consequently, to protect confidentiality, redundancy may need to combined with, e.g., secret sharing schemes [25, 46].

We can model redundancy by allowing the defender to choose the set of deployed components C from a family $\mathscr{C}$ of feasible sets. This family $\mathscr{C}$ consists of all the sets that are sufficient for providing required system functionality and performance. In a simple model, we may assume that a base deployment C_{base} is given, and the defender can choose only supersets $C \supseteq C_{base}$ (i.e., $\mathscr{C} = \{C \mid C \supseteq C_{base}\}$). By deploying additional components, the defender incurs some cost. In the case of hardware, this is the cost of purchasing, installing, and operating devices, which may be an additive or submodular function of the set of additional devices (i.e., fixed cost or diminishing marginal cost model). In the case of services and software modules, this may be development cost or the computational/communication cost of running an additional software components.

6.5.2 Diversity

Deploying redundant components may be a futile effort if all of the components are implemented and configured in the same way since an attacker might be able to compromise all of them with relatively little effort using a common software or configuration vulnerability. A defender can prevent this by implementing the components using a *diverse set of hardware and software*, for example, by running redundant web servers on different operating systems. Diversity reduces the impact of any common vulnerability since only the components that are implemented using the vulnerable software or hardware will be susceptible to the same exploit. Indeed, diversity has been recognized as an effective approach for improving network security, and prior work has studied the optimal assignment of implementation types to components [42]. On a larger, societal scale, monoculture (i.e., lack of diversity in software solutions) has been identified as a contributor to systemic cyber-risks [6, 18].

Similar to redundancy, diversity must be used carefully since it may increase risks in some cases. If an attacker needs to compromise a certain set of components to inflict damage, then diversity increases resilience since the probability of finding a vulnerability in multiple implementations is generally lower than finding one in a single implementation. However, if the attacker needs to compromise only one out of many components, then diversity decreases resilience since the more implementation types, the higher the probability that at least one of them has a vulnerability.

We can model diversity by letting the defender assign an implementation type to each component. More specifically, for each component $c \in C$, we can assume that a set of feasible implementations I_c is given, and the defender can select a particular implementation i_c. In practice, the defender typically incurs some cost for introducing a new implementation type into the system. For example, introducing a new software may require purchasing licenses and training for personnel. Consequently, the cost of diversity depends on the set of all the implementation types $\bigcup_{c \in C}\{i_c\}$ that are used in the system.

6.5.3 Isolation

While links serve a useful purpose by providing connectivity between components, they also enable an attacker to escalate its attack by compromising the neighbors of a compromised component. A defender can prevent escalation and limit the impact of compromises by *isolating* components (or sets of components) from each other [48]. In practice, techniques for isolation range from sandboxes for software components to firewalls between networks. To minimize security risks, isolation may be implemented on a physical level by introducing an “air gap” (i.e., physical separation) between components. In the context of cyber-physical systems, “air gap” is typically used to protect safety-critical control systems [15].

We can model isolation by allowing the defender to remove links from the network. Equivalently, we may allow the defender to choose the set of links E to retain, under the constraint that this set of links must be chosen from a family $\mathscr{E}$ of feasible sets. This family $\mathscr{E}$ consists of all the sets that are sufficient for providing connectivity that is necessary for the required system functionality and performance. We may define the family of feasible sets using graph-theoretic notions; for example, we may require the set of links E to form a strongly connected graph of components C.

By severing useful links between components, the defender incurs various costs. For example, decreased connectivity may result in lower performance or functionality as well as increased usability and operational costs (e.g., information that could have been sent automatically on a link might have to be transferred manually using removable drives). Consequently, a defender must carefully choose which components to isolate from each other. Dividing a networked system into isolated parts is a challenging graph-theoretic problem, which has been studied in prior work, e.g., as a computationally-hard graph partitioning problem [4].

6.5.4 Hardening

Hardening includes techniques for protecting components from being compromised by an attacker. These techniques may be applied at either the hardware level (e.g., using tamper-resistant hardware to prevent attacks based on physical access) or at the software level (e.g., thorough testing for software vulnerabilities). Typically, hardware-level techniques are applied to individual components (i.e., protection for particular devices), while software-level techniques are applied to a set of components that are implemented using the same software (i.e., eliminating common vulnerabilities). Defenders may employ a variety of approaches for eliminating software vulnerabilities, ranging from following secure-coding principles to hiring outside experts for penetration testing or crowdsourcing vulnerability discovery through bug-bounty programs [36, 58].

For hardware-level protection, we can model hardening by allowing the defender to choose how much to spend on improving the security of each component. For software-level protection, the defender needs to choose how much to spend on improving certain implementation types $i \in I$, where $I = \cup_{c \in C} I_c$ is the set of all implementation types in the system. In both cases, hardening either decreases the probability that an attack succeeds against the hardened components, or it increases the cost of launching a successful attack against the hardened components. Optimal security investments have been thoroughly studied in the economics of security literature [3, 22].

6.5.5 Detection

With respect to information, the defender is at a grave disadvantage compared to the attacker. Since the defender has imperfect information regarding which components have been attacked or compromised, it can only guess which actions to take to mitigate a potential attack most effectively. To decrease this information gap, defenders can deploy *intrusion detection* systems. An intrusion detection system (IDS) monitors a system or network and raises an alarm when it encounters malicious activity, which can then be investigated by system operators. In practice, IDS come in a wide variety. A host-based IDS is deployed on and monitors a particular host (e.g., running processes), while a network-based IDS monitors network traffic. A signature-based IDS searches for known attacks, while an anomaly-based IDS looks for deviation from normal operation. A variety of intrusion detection systems have also been proposed for cyber-physical systems [40]

However, practical intrusion detection systems are imperfect. On the one hand, they may fail to detect an actual attack, which is called a false-negative error. On the other hand, they may raise an alarm when they encounter suspicious but non-malicious activity, which is called a false-positive error. Both of these are errors should be minimized since false negatives prevent the defender from mitigating attacks, while false positives waste the limited amount of time and effort available for investigations. However, there is generally a trade-off between the two errors: decreasing the rate of false positives results in an increased rate of false negatives, and vice versa. Therefore, defender must carefully configure each IDS to minimize losses due to attacks and the costs of investigations at the same time. Finding optimal configurations for intrusion detection systems is a challenging problem by itself [19, 20, 34].

We can model detection by allowing the defender to place intrusion detection systems on components or links. We let $S^C \subseteq C$ denote the set of components with detectors, which model host-based IDS, and let $S^E \subseteq E$ denote the set of links with detectors, which model network-based IDS. For each IDS $s \in S^C \cup S^E$, the defender must choose a trade-off between false-negative and false-positive errors by configuring the detector. We can represent the attainable combinations using a trade-off function $F_s : \mathbb{R}_+ \times A \rightarrow [0, 1]$, where $F_s(f_s, a)$ is the estimated probability that attack a is undetected when the false-positive error rate of the detector is f_s. In each timestep, the defender's beliefs are updated based on which detectors have raise a true alarm, while the defender incurs cost for all the false alarms raised $\sum_{s \in S^C \cup S^E} f_s$.

6.6 Reactive Defense

Reactive defense includes actions taken by a defender in response to observed attacks. Here, we discuss approaches for the reactive defense of networked systems in more detail, focusing on how to incorporate them into our game-theoretic model. In contrast to proactive defense, these actions are taken in time steps $t > 0$ (i.e., after an attack may have been launched).

6.6.1 Islanding

Isolation can be an effective approach for limiting the impact of successful attacks (Sect. 6.5.3); however, it requires severing links proactively, which results in permanent usability and performance degradation, even when the defender has not observed an attack. Here, we consider *islanding*, which can be thought of as a reactive variant of isolation that severs links only after detecting an attack. More specifically, islanding means severing links to components (or to a set of components) that the defender suspects to be compromised by an attacker.

Islanding clearly has some advantages over isolation, but it can also be favorable compared to simply shutting down and resetting (e.g., re-installing) components that are suspected to be compromised. Since the defender possesses imperfect information regarding which components have been compromised, implementing any reactive defense is risky in the sense that the actions might not only be costly but also unnecessary. Shutting down and resetting a component is a drastic measure that may result in significant downtime. In a cyber-physical system, such as a power plant, where components have to sense and control physical processes in real time, downtime can be prohibitively expensive. On the other hand, islanding allows the defender to prevent the escalation of a suspected attack without shutting down the potentially compromised components. If these islanded components can provide some level of functionality (e.g., an islanded component in a cyber-physical system may still be able to control a physical process), then islanding can be a less risky option for the defender.

We can model islanding similar to isolation, by allowing the defender to choose which links are active in each time step. More formally, in each time step t, the defender may choose a set of active links $E_t \subseteq E$. Then, the attacker will only be able use links E_t to escalate its attack in time step t, while the defender incurs cost due to the performance and usability loss from the unavailability of links $E \setminus E_t$.

6.6.2 Resetting

Even though islanding can contain a security breach by preventing the attacker from escalating the attack to compromise other components, it cannot eliminate the breach and secure the system. We now consider actions that return compromised components into their normal, uncompromised state, to which we refer as *resetting*. For components that model physical hosts, resetting typically involves shutting down and re-installing the hosts, while for software components, re-launching running processes may be enough to bring them into a secure state as long as they have not effected any permanent changes to, e.g., configuration files.

By resetting a component, the defender incurs cost due to the effort and time required to reset the component, as well as the cost of the component being unavailable while it is being reset. Since the defender does not have perfect information, it does not know when to reset a component: resetting a component that has not been compromised results in unnecessary expenses, while not resetting a compromised one may result in increased losses due to the prolonged impact of the attack. Consequently, deciding when to reset a potentially compromised component is a challenging problem. This problem has been studied extensively by prior work using the FlipIt model [7, 31, 33, 51, 57], and optimal resetting schedules have been proposed under various conditions. However, integrating these results into a multi-stage game where a variety of actions are available to the defender is an open problem, especially considering the structural properties of networked systems.

We can model resetting by allowing the defender to select which components $R_t \subseteq C$ to reset in each time step t. Selected components R_t are removed from the sets of compromised and impaired components C_t^C and C_t^I, respectively, but they also become (or remain) unavailable for a certain number of time steps, which models downtime due to resetting. Further, the defender may incur two types of costs. Firstly, it incurs the direct cost of resetting the components, which can be modeled as an additive function of R_t. Secondly, it incurs the cost of lost performance or functionality due to the downtime of the selected components, which may depend on the deployment of the system. For example, if there are redundant components available, there might not be any performance or functionality loss.

6.6.3 Reconfiguration

In addition to islanding and resetting potentially compromised components, the defender may also mitigate attacks and limit their impact by changing the behavior of uncompromised components. In particular, the defender can reconfigure components that are still available and under its control in order to reduce the losses arising from an attack. For example, in a cyber-physical system, a controller may be reconfigured when some sensors or actuators are compromised or impaired, so that the new control maintains system stability and prevents system failure in spite of the attack [8].

We can model reconfiguration actions by letting the defender select in every time step a configuration for each available component. Formally, in each time step t, the defender selects for each component $c \in C \setminus C_t^I$ a configuration $F_{c,t}$. However, the configurations are applied only to uncompromised components $(C \setminus C_t^I) \setminus C_t^C$ (note that the defender does not necessarily know which components are compromised and which are under its control). Reconfiguring a component c may have some cost, such as the effort exerted to effect the change or the loss due to temporary outage while reconfiguring components, which the defender incurs only if it actually changes the configuration, i.e., if $F_{c,t} \neq F_{c,t-1}$. Further, the selected configurations may also have an impact on the performance and functionality of the system (e.g., in a cyber-physical system, a more stable controller may be less efficient), which affects the defender's utility.

6.7 Solving Multi-Stage Security Games

Our goal is to find an optimal defense policy π_D, which proscribes what defensive actions to take in each time step based on the observed state. Unfortunately, this problem is computationally challenging due to the sizes of the action and state spaces. Firstly, in each time step, the defender has to choose from a set of actions whose cardinality is an exponential function of the size of the graph (C, E) that models the system. For example, consider isolation and islanding actions, which are chosen from the set of all subsets of links E. Since the number of all subsets is $2^{|E|}$, the size of the defender's action space can be astronomical even for graphs of modest size.[7] Similarly, the number of possible states is also an exponential function

[7] Some of these subsets may not be feasible, but in general the number of feasible subsets may grow exponentially.

of the size of the graph (C, E). For instance, the set of compromised components C^C is a subset of the set of components C; hence, there may be up to $2^{|C|}$ different sets of compromised components. Further, the set of possible observations for the defender, which may include various alerts generated by detectors, could be even larger.

Considering the sizes of the action and state spaces, it is challenging not only to find an optimal policy, but even just to represent one. A straightforward policy representation would specify what actions to take for each possible state, for example, in the form of a list. Clearly, the size of this list would be prohibitively large for any practical system. Therefore, there is a need for devising a *compact representation* of proactive and reactive action policies.

Even restricted to some compact representation, finding an optimal policy may be computationally hard. Indeed, prior work has shown that a number of subproblems (e.g., finding optimal actions of a certain type in a given state) are NP-hard. For example, finding optimal configurations for intrusion detection systems may be an NP-hard problem when facing strategic attacks [19, 34]. In light of this, we must consider *efficient algorithms* that can find near-optimal actions. To devise such algorithms, we can take advantage of the structure of our problem. In other words, instead of resorting to generic meta-heuristics, we can tailor our algorithms to the rich structure of security states and defensive actions.

We can combine these algorithms with reinforcement learning approaches for finding near-optimal policies [41]. Many reinforcement learning algorithms, such as Q-learning [54], work by learning the values of the possible states (e.g., a state in which more components are compromised may be worth less to the defender than a state with fewer compromised components). Once these values have been learned, the best action in a certain state can be chosen based on which action results in the highest expected value for the following state, considering the probabilities of the various state transitions for a particular action.

Since exhaustively searching for the best action would not be feasible in our model, we propose to use an actor-critic method [29], which represents both the state values and the policy explicitly. Considering the complexity of the state and action spaces, there is a need to represent the state values and the policy efficiently, which we may do using (deep) neural networks. This model-free approach can be combined with efficient, model-specific algorithms for finding a near-optimal action in a particular state to support the exploration part of reinforcement learning.

However, considerable challenges remain in the application of reinforcement learning to multi-stage security games. Firstly, the actor-critic method can be used directly to find a near-optimal policy for one player, which constitutes an approximate best response, against a given policy of the opponent. Solving the game and finding a strategic cyber-defense policy, however, requires finding an equilibrium pair of defender and attacker policies. To find a mixed-strategy equilibrium, we can apply a double-oracle approach, which starts with a restricted set of strategies (i.e., policies), and then iteratively computes a mixed equilibrium over the restricted set and extends the set with best-response strategies against this equilibrium [39]. The application of a double-oracle approach may lead to further computational problems since computing many best-response strategies (i.e., running reinforcement learning many times to find policies) can be computationally expensive.

Another challenge arises from the fact that the players in our game have neither complete nor perfect information. Consequently, reinforcement learning has to find near-optimal policies for partially observable Markov decision processes. To achieve good results for partially observable processes, we can extend the actor-critic method with an internal state, for example, using recurrent neural networks [24].

6.8 Conclusion

To protect sensitive networked systems, defenders need to deploy complex cyber-defense solutions, which combine a variety of proactive and reactive techniques to minimize cyber-risks. Devising complex defense solutions for practical systems is a daunting task, which must be supported by strong theoretical models and efficient tools. To address this need, we introduced a modeling framework for high-resolution multi-stage security games for networked systems. We discussed a number of canonical proactive and reactive defense approaches, focusing on modeling choices and challenges. Finally, we considered the computational problem of finding optimal defense policies.

There remain several open problems in the area of high-resolution multi-stage security games. While we have laid foundations for theoretical models, incorporating a spectrum of practical defense methods into this framework requires further modeling work. Then, models need to be rigorously evaluated using data regarding past cyber-breaches as well as the architecture, performance, and functionality of a wide range of practical networked systems. Once these models have been established, the gap between theory and practice must be bridged by providing software tools for practitioners that facilitate the application of models to practical systems. Finally, finding optimal defense policies poses a very challenging computational problem. We have outlined approaches for addressing this problem, but developing efficient practical algorithms and tailoring reinforcement learning methods to multi-stage security games remain open problems.

References

1. W. Abbas, A. Laszka, X. Koutsoukos, Resilient wireless sensor networks for cyber-physical systems, in *Cyber-Physical System Design with Sensor Networking Technologies*, ed. by S. Zeadally, N. Jabeur (The Institution of Engineering and Technology, Stevenage, 2016), pp. 239–267
2. M. Abrams, J. Weiss, Malicious control system cyber security attack case study – Maroochy Water Services, Australia (2008). http://csrc.nist.gov/groups/SMA/fisma/ics/documents/Maroochy-Water-Services-Case-Study_report.pdf
3. R. Anderson, T. Moore, The economics of information security. Science **314**(5799), 610–613 (2006)

4. J. Aspnes, K. Chang, A. Yampolskiy, Inoculation strategies for victims of viruses and the sum-of-squares partition problem. J. Comput. Syst. Sci. **72**(6), 1077–1093 (2006)
5. A. Avizienis, J.C.Laprie, B. Randell, C. Landwehr, Basic concepts and taxonomy of dependable and secure computing. IEEE Trans. Dependable Secure Comput. **1**(1), 11–33 (2004)
6. K.P. Birman, F.B. Schneider, The monoculture risk put into context. IEEE Secur. Privacy **7**(1), 14–17 (2009)
7. 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, in *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)* (Springer, New York, 2012), pp. 248–263
8. L.F. Cómbita, J. Giraldo, A.A. Cárdenas, N. Quijano, Response and reconfiguration of cyber-physical control systems: a survey, in *Proceedings of the 2nd Colombian Conference on Automatic Control (CCAC)* (IEEE, New York, 2015), pp. 1–6
9. Council of Economic Advisers, The cost of malicious cyber activity to the U.S. economy. Tech. rep., Executive Office of the President (2018)
10. C.F. Daganzo, The cell transmission model: a dynamic representation of highway traffic consistent with the hydrodynamic theory. Transp. Res. Part B Methodol. **28**(4), 269–287 (1994)
11. C.F. Daganzo, The cell transmission model, part II: network traffic. Transp. Res. Part B Methodol. **29**(2), 79–93 (1995)
12. D. Dasgupta, A. Roy, A. Nag, Multi-factor authentication, in *Advances in User Authentication* (Springer, New York, 2017), pp. 185–233
13. L. Dritsoula, P. Loiseau, J. Musacchio, Computing the nash equilibria of intruder classification games, in *International Conference on Decision and Game Theory for Security* (Springer, New York, 2012), pp. 78–97
14. L. Dritsoula, P. Loiseau, J. Musacchio, A game-theoretic analysis of adversarial classification. IEEE Trans. Inf. Forensics Secur. **12**(12), 3094–3109 (2017)
15. J.P. Farwell, R. Rohozinski, Stuxnet and the future of cyber war. Survival **53**(1), 23–40 (2011)
16. J. Filar, K. Vrieze, *Competitive Markov Decision Processes* (Springer, New York, 1997)
17. GE Digital, The impact of cyber attacks on critical infrastructure. Tech. rep., General Electric (2017)
18. D. Geer, R. Bace, P. Gutmann, P. Metzger, C. Pfleeger, J. Querterman, B. Scheier, CyberInsecurity: the cost of monopoly–how the dominance of Microsoft's products poses a risk to security. Tech. rep., Computer and Communications Industry Association (2003)
19. A. Ghafouri, W. Abbas, A. Laszka, Y. Vorobeychik, X. Koutsoukos, Optimal thresholds for anomaly-based intrusion detection in dynamical environments, in *Proceedings of the 7th Conference on Decision and Game Theory for Security (GameSec)* (2016), pp. 415–434
20. A. Ghafouri, A. Laszka, A. Dubey, X. Koutsoukos, Optimal detection of faulty traffic sensors used in route planning. in *Proceedings of the 2nd International Workshop on Science of Smart City Operations and Platforms Engineering (SCOPE)* (2017), pp. 1–6
21. B. Ghena, W. Beyer, A. Hillaker, J. Pevarnek, J.A. Halderman, Green lights forever: analyzing the security of traffic infrastructure. in *Proceedings of the 8th USENIX Workshop on Offensive Technologies (WOOT)*, vol. 14 (2014), pp. 1–10
22. L.A. Gordon, M.P. Loeb, The economics of information security investment. ACM Trans. Inf. Syst. Secur. **5**(4), 438–457 (2002)
23. S. Grad, Engineers who hacked into LA traffic signal computer, jamming streets, sentenced. Los Angeles Times (2009)
24. M. Hausknecht, P. Stone, Deep recurrent Q-learning for partially observable MDPs, in *2015 AAAI Fall Symposium Series* (2015)
25. E. Karnin, J. Greene, M. Hellman, On secret sharing systems. IEEE Trans. Inf. Theory **29**(1), 35–41 (1983)
26. Kaspersky Labs' Global Research & Analysis Team, Gauss: abnormal distribution (2012). https://securelist.com/analysis/36620/gauss-abnormal-distribution/

27. M.B. Kelley, The Stuxnet attack on Iran's nuclear plant was 'far more dangerous' than previously thought. *Business Insider* (2013). http://www.businessinsider.com/stuxnet-was-far-more-dangerous-than-previous-thought-2013-11
28. A. Kerckhoffs, La cryptographie militaire. J. Sci. Mil. **IX**, 5–83 (1883)
29. V.R. Konda, J.N. Tsitsiklis, Actor-citic agorithms, in *Proceedings of the 12th International Conference on Neural Information Processing Systems (NIPS)* (MIT Press, Cambridge, 1999), pp. 1008–1014
30. D. Korzhyk, Z. Yin, C. Kiekintveld, V. Conitzer, M. Tambe, Stackelberg vs. nash in security games: an extended investigation of interchangeability, equivalence, and uniqueness. J. Artif. Intell. Res. **41**, 297–327 (2011)
31. A. Laszka, B. Johnson, J. Grossklags, Mitigating covert compromises: a game-theoretic model of targeted and non-targeted covert attacks, in *Proceedings of the 9th Conference on Web and Internet Economics (WINE)* (2013), pp. 319–332
32. A. Laszka, M. Felegyhazi, L. Buttyan, A survey of interdependent information security games. ACM Comput. Surv. **47**(2), 23:1–23:38 (2014)
33. A. Laszka, G. Horvath, M. Felegyhazi, L. Buttyan, FlipThem: modeling targeted attacks with FlipIt for multiple resources, in *Proceedings of the 5th Conference on Decision and Game Theory for Security (GameSec)* (2014), pp. 175–194
34. A. Laszka, W. Abbas, S.S. Sastry, Y. Vorobeychik, X. Koutsoukos, Optimal thresholds for intrusion detection systems, in *Proceedings of the 3rd Annual Symposium and Bootcamp on the Science of Security (HotSoS)* (2016), pp. 72–81
35. A. Laszka, B. Potteiger, Y. Vorobeychik, S. Amin, X. Koutsoukos, Vulnerability of transportation networks to traffic-signal tampering, in *Proceedings of the 7th International Conference on Cyber-Physical Systems (ICCPS)*, p. 16 (IEEE Press, Piscataway, 2016)
36. A. Laszka, M. Zhao, J. Grossklags, Banishing misaligned incentives for validating reports in bug-bounty platforms, in *Proceedings of the 21st European Symposium on Research in Computer Security (ESORICS)*, pp. 161–178 (2016)
37. R.M. Lee, M.J. Assante, T. Conway, Analysis of the cyber attack on the Ukrainian power grid: defense use case. Tech. rep., Electricity Information Sharing and Analysis Center (E-ISAC) (2016)
38. 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)
39. H.B. Mcmahan, G.J. Gordon, A. Blum, Planning in the presence of cost functions controlled by an adversary, in *International Conference on Machine Learning* (2003), pp. 536–543
40. R. Mitchell, I.R. Chen, A survey of intrusion detection techniques for cyber-physical systems. ACM Comput. Surv. **46**(4), 55 (2014)
41. V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, et al., Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
42. A.J. O'Donnell, H. Sethu, On achieving software diversity for improved network security using distributed coloring algorithms, in *Proceedings of the 11th ACM Conference on Computer and Communications Security (CCS)* (ACM, New York, 2004), pp. 121–131
43. R. Píbil, V. Lisý, C. Kiekintveld, B. Bošanský, M. Pěchouček, Game theoretic model of strategic honeypot selection in computer networks, in *Proceedings of the 3rd Conference on Decision and Game Theory for Security (GameSec)*, (Springer, New York, 2012), pp. 201–220
44. P. Polityuk, Ukraine investigates suspected cyber attack on Kiev power grid. Reuters (2016). http://www.reuters.com/article/us-ukraine-crisis-cyber-attacks-idUSKBN1491ZF
45. Ponemon Institute, 2017 cost of data breach study. Tech. rep., IBM, 2017
46. T. Rabin, M. Ben-Or, Verifiable secret sharing and multiparty protocols with honest majority, in *Proceedings of the 21st Annual ACM Symposium on Theory of Computing (STOC)* (ACM, New York, 1989), pp. 73–85

47. A. Schlenker, O. Thakoor, H. Xu, M. Tambe, P. Vayanos, F. Fang, L. Tran-Thanh, Y. Vorobeychik, Deceiving cyber adversaries: a game theoretic approach, in *Proceedings of the 17th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)* (2018)
48. R. Shu, P. Wang, S.A. Gorski III, B. Andow, A. Nadkarni, L. Deshotels, J. Gionta, W. Enck, X. Gu, A study of security isolation techniques. ACM Comput. Surv. (CSUR) **49**(3), 50 (2016)
49. J. Slay, M. Miller, Lessons learned from the Maroochy water breach. in *Critical Infrastructure Protection*, ed. by E. Goetz, S. Shenoi (Springer, New York, 2008), pp. 73–82
50. M. Tambe (ed.), *Security and Game Theory: Algorithms, Deployed Systems, Lessons Learned* (Cambridge University Press, Cambridge, 2011)
51. M. van Dijk, A. Juels, A. Oprea, R.L. Rivest, FlipIt: the game of "stealthy takeover". J. Cryptol. **26**(4), 655–713 (2013)
52. Y. Vorobeychik, S. Singh, Computing stackelberg equilibria in discounted stochastic games, in *National Conference on Artificial Intelligence* (2012)
53. Y. Vorobeychik, B. An, M. Tambe, S. Singh, Computing solutions in infinite-horizon discounted adversarial patrolling games, in *International Conference on Automated Planning and Scheduling* (2014)
54. C.J. Watkins, P. Dayan, Q-learning. Mach. Learn. **8**(3-4), 279–292 (1992)
55. K. Zetter, Hackers can mess with traffic lights to jam roads and reroute cars. WIRED (2014). https://www.wired.com/2014/04/traffic-lights-hacking/
56. K. Zetter, Inside the cunning, unprecedented hack of Ukraine's power grid. WIRED (2016). https://www.wired.com/2016/03/inside-cunning-unprecedented-hack-ukraines-power-grid/
57. M. Zhang, Z. Zheng, N.B. Shroff, A game theoretic model for defending against stealthy attacks with limited resources, in *Proceedings of the 6th Conference on Decision and Game Theory for Security (GameSec)* (Springer, New York, 2015), pp. 93–112
58. M. Zhao, J. Grossklags, P. Liu, An empirical study of web vulnerability discovery ecosystems, in *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS)* (ACM, New York, 2015), pp. 1105–1117

Chapter 7
Moving Target Defense for Attack Mitigation in Multi-Vehicle Systems

Jairo Giraldo and Alvaro A. Cardenas

Abstract Cyber-Physical Systems (CPS) have traditionally been considered more static with more regular communication patterns when compared to classical information technology networks. Because the structure of most CPS remains unchanged during long periods of times, they become vulnerable to adversaries with the precise knowledge of the system, and who can tailor their attacks based on their knowledge about the system dynamics, communications, and control.

Moving Target Defense (MTD) has emerged as a key strategy to add uncertainty about the state and execution of a system in order to prevent attackers from having predictable effects with their attacks. In the last few years MTD has been used in different CPS scenarios by adding uncertainties into the physical characteristics of the system. Most of these applications are used to detect attacks, or to make difficult for attackers to gather information. In this chapter, we propose an MTD strategy for multi-vehicle systems that can be used to mitigate the impact caused by cyber-attacks. We characterize the trade-off between impact mitigation and performance degradation, and illustrate the viability of our approach in two applications, (1) vehicular platooning, and (2) UAV formation. Finally, we extend our results to a more general control systems framework, and we introduce different types of MTD mechanisms, i.e., at the controller level and at sensors.

J. Giraldo
Erik Jonsson School of Engineering, University of Texas at Dallas, Richardson, TX, USA
e-mail: jairo.giraldo@utdallas.edu

A. A. Cardenas (✉)
Erik Jonsson School of Engineering, University of Texas at Dallas, Richardson, TX, USA

Baskin School of Engineering, University of California, Santa Cruz, Santa Cruz, CA, USA
e-mail: alvaro.cardenas@ucsc.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances
in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_7

7.1 Introduction

Moving target defense (MTD) has been proposed as a way to make difficult the reliable exploitation of a system by attackers because it makes the attack surface dynamic [5]. For instance Dunlop et al. [3] proposed MT6D, an MTD mechanism for IPv6 which maintains user privacy and protects against targeted network attacks by repeatedly rotating the addresses of both the sender and receiver. Similarly, Wang et al. [22] introduced MOTAG, a strategy that defends against Internet DDoS attacks, by employing a layer of secret random proxy nodes to relay communications between clients and the protected application servers.

Most applications of MTD have been used for network protection and to secure applications. However, in the last couple of years the use of MTD techniques has been extended to protect cyber-physical systems. Several authors have used MTD approaches for state estimation in the smart grids [2, 16, 20], where the main idea consists on changing the physical topology of the power grid in order to reveal false data injection attacks. Weerakkody and Sinopoli [23] proposed the addition of an external system unknown to the attacker that uses additional sensor readings to obtain an estimate, making it harder for an adversary to design stealthy attacks. A similar approach was introduced by Valente and Cárdenas [21], where external visual challenges (e.g., a screen with extra information) are used to verify the authenticity of video footage. Closer to our work, Pang et al. [12] considered DDoS attacks that can shut down control commands; to prevent this attack, they propose the use of multiple distributed controllers so when a control command is not received, another controller is selected. On the other hand, Kanellopoulos and Vamvoudakis [6] propose a proactive MTD mechanism that consists on randomly switching among multiple controllers to increase the unpredictability of the control system. The switching probabilities are selected in order to maximize the entropy produced by the switching strategy while ensuring minimum controller cost. One of our approaches is similar, but we focus on minimizing the impact of the attack instead of the entropy. However, in our formulation it is possible to include the entropy maximization as an additional objective.

In this chapter we show how MTD can be used not only to increase the cost and difficulty of designing cyber-attacks, but also to mitigate the impact of successful attacks. We propose the use of random communication topologies for multi-vehicle systems as a moving target mechanism that can be designed to decrease the negative impact of the attack. We derive stability conditions for second-order consensus protocols in the presence of random switching topologies and we identify trade-offs between the convergence rate and the attack impact. The viability of our approach is illustrated with two case studies, (1) vehicular platooning, where a group of vehicles need to remain close enough to exploit the benefits of the platoon (i.e., decreasing CO2 emissions and fuel consumption) while avoiding collisions, and (2) Unmanned Aerial Vehicle (UAV) formation, where a group of UAVs need to maintain a formation that can be used for surveillance or exploration. Finally, we extend our analysis to a more general framework and introduce novel MTD strategies that induce random switching between different controllers, or between sensors.

We formulate optimization problems in order to obtain the optimal probability distribution that minimizes the impact of the attack.

Preliminaries and Notation

Graph theory: Let $\mathscr{G} = (\mathscr{V}, \mathscr{E}, \mathscr{A})$ represents a graph, where $\mathscr{V} = \{1, 2, \ldots, N\}$ is the set of nodes or vertices, and $\mathscr{E}\{(i, j)|i, j \in \mathscr{V}\}$ is the set of pairs called edges. If a pair $(i, j) \in \mathscr{E}$, then i, j are adjacent. The adjacency matrix $\mathscr{A} = [a_{ij}]$ is the symmetric (nonsymmetric for directed graphs) matrix $N \times N$, where $a_{ij} = 1$ if (i, j) are adjacent, $a_{ij} = 0$ otherwise. For the ith node, the set of neighbors is $N_i = \{j|(i, j) \in \mathscr{E}\}$, and the degree of a vertex d_i^s is the number of neighbors that are adjacent to i, i.e., $d_i^s = \sum_{j=1}^{N} a_{ij}$ or, for directed graphs, the number of neighbors whose direction is heading to node i. A sequence of edges (i_1, i_2), (i_2, i_3), $\ldots$, (i_{r-1}, i_r) is called a path from node i_1 to node i_r. The graph $\mathscr{G}$ is said to be connected if for any $i, j \in \mathscr{V}$ there is a path from i to j. The degree matrix is $\mathscr{D} = diag(d_1, d_2, \ldots, d_N)$, and the Laplacian of $\mathscr{G}$ is defined as $\mathscr{L} = \mathscr{D} - \mathscr{A}$. A graph is said to be a k-regular graph if all vertices have connectivity equal to k, each node is connected to k neighbors.

7.2 MTD for Multi-Agent Systems

Multi-agent systems (MAS) are systems that capture a variety of social and distributed interactions where agents make decisions based only on local information (See Fig. 7.1). One of the main components of MAS are the communication links, that indicate whether or not one agent shares information with another. Unfortu-

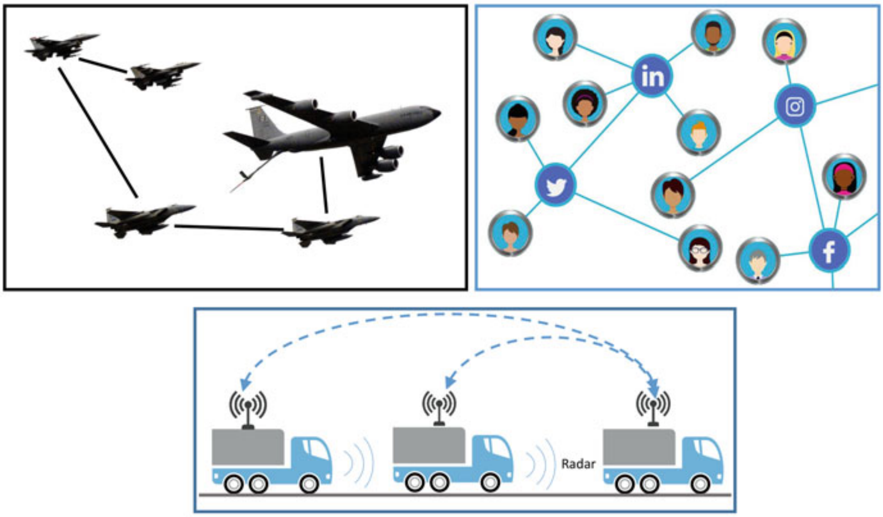


Fig. 7.1 Examples of multi-agent systems

nately, MAS are susceptible to adversaries that may gain access to a subset of communication links and inject false information. For instance, a man-in-the-middle attack can inject false data about a specific sensor, or in social networks, releasing false information to a subset of people in a group that interacts to complete a specific task. In this chapter, we propose MTD strategies to help to mitigate the effects of false data injection attacks in MAS, with emphasis on multi-vehicle systems.

Second-Order Multi-Vehicle System

Let us consider a system with n agents that update their states using the information from a set of neighbors. Each agent is represented by a discrete-time second-order integrator of the form

$$\begin{aligned} x_i(k+1) &= x_i(k) + v_i(k) \\ v_i(k+1) &= v_i(k) + u_i(k), \quad \text{for all } i \in \ell = \{1, 2, \ldots, n\}. \end{aligned} \tag{7.1}$$

where $x_i(k) \in \mathbb{R}$ and $v_i(k) \in \mathbb{R}$ are the position and velocity of each agent i at time k, respectively. Typically, a distributed control action $u_i(k)$ is designed by considering information from a set of neighbors. The communication interaction among agents is modeled by a time-varying directed graph $\mathscr{G}(k) = (\mathscr{V}, \mathscr{E}(k), \mathscr{A}(k))$, where each vertex represent an agent, and the set of communication links are described by $\mathscr{E}(k)$, where the link $e_{ij}(k) \in \mathscr{E}(k)$ if node i receives information from j. Therefore, we consider the consensus protocol adapted from [25] with dynamic communication interactions described by

$$\begin{aligned} u_i(k) = &- \alpha_1 \sum_{j=1}^{n} a_{ij}(k)(x_i(k) - x_j(k) - \delta_{ij}(k)) \\ &- \alpha_2 \sum_{j=1}^{n} a_{ij}(k)(v_i(k) - v_j(k) - \gamma_{ij}(k)) \end{aligned} \tag{7.2}$$

where $a_{ij}(k)$ are the elements of the time-varying adjacency matrix $\mathscr{A}(k)$, α_1, α_2 are parameters to be designed, and $\delta_{ij}(k)$, $\gamma_{ij}(k)$ correspond the attack injected in the information that agent i receives from its neighbor j, for $\delta_{ij}(k) \neq \delta_{ji}(k)$, and $\gamma_{ij}(k) \neq \gamma_{ji}(k)$.

Attacker Model

We consider an adversary that has knowledge about the system dynamics and parameters α_1, α_2, and he knows the fixed communication topology that represents all possible communications. Let k_a, k_f denote the initial and final time of the attack. Thus, the adversary can craft the attack sequences $\{\phi(k_a), \phi(k_a+1), \ldots, \phi(k_f)\}$ and $\{\gamma(k_a), \ldots, u(k_fa)\}$. We assume that an adversary is able to hijack a subset of communication links and modify the information sent from agent i to agent j. This model may represent two types of attacks as depicted in Fig. 7.2: Sybil attack, where an adversary falsifies the identity of an agent and starts sending false information;

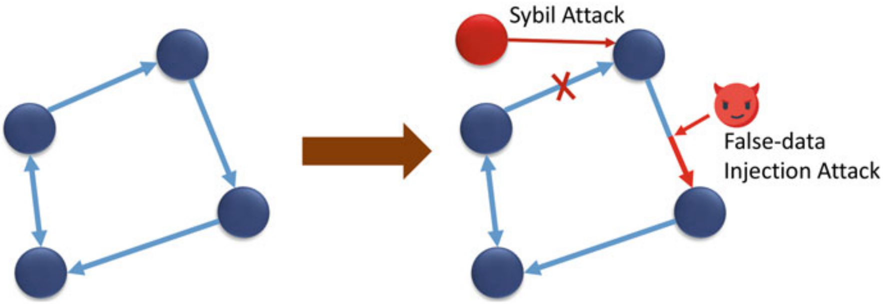


Fig. 7.2 Example of two types of attacks considered in this chapter

and false-data injection attacks, where the attacker intercepts the communications between two agents and falsify the information that is being transmitted. We do not assume that a sensor is compromised, but only the communication channel used to transmit the sensor information to a specific neighbor. For instance, for agents 1, 2, 3, the adversary may compromise the information of y_1 sent from 1 to 2, but not the information of y_1 sent from 1 to 3.

7.2.1 Random Communication Topology

The use of random communication topologies for first-order consensus algorithm are useful to model uncertainties in the system such as link failures or DDoS attacks [7, 13]. In this work, we propose the use of random topologies as an MTD strategy that can help to mitigate the impact of adversaries. In particular, we focus on the second-order consensus algorithm in Eq. (7.2) and we derive sufficient conditions for stability.

Let us define the total graph (or supergraph) $\mathcal{G}_T = (\mathcal{V}, \mathcal{E}_T, \mathcal{A}_T)$ as the fixed graph that represents *all possible* communications between agents, where the set $\mathcal{E}_T$ collects all the channels that can be established directly among pairs of sensors, i.e., it is the set of realizable edges. Without an MTD policy, we consider that the communication topology is represented by a fixed graph $\mathcal{G}_f$, which is a spanning connected subgraph of $\mathcal{G}_T$, such that $\mathcal{E}_f \subseteq \mathcal{E}_T$.

Now, our MTD strategy can be modeled by the time-varying graph $\mathcal{G}(k) = (\mathcal{V}, \mathcal{E}(k), \mathcal{A}(k))$ with fixed vertex set $\mathcal{V}$, and time-varying edge set $\mathcal{E}(k) \subset \mathcal{E}_T$, where the edges can vary with time either deterministically or completely random. The instantaneous Laplacian matrix is then $L(k)$.

Now, let $x(k) = [x_1(k),\ x_2(k),\ \ldots,\ x_n(k)]^\top$, $v(k) = [v_1(k),\ \ldots,\ v_n(k)]^\top$, and $z(k) = [x(k)^\top,\ v(k)^\top]$. Also, let $\boldsymbol{\delta_i}(k) = \sum_{j=1}^{n} a_{ij}(k)\delta_{ij}(k)$ and $\boldsymbol{\gamma_i}(k) = \sum_{j=1}^{n} a_{ij}(k)\gamma_{ij}(k)$ and $\boldsymbol{\delta}(k) = [\boldsymbol{\delta}_1(k),\ \ldots,\ \boldsymbol{\delta}_n]^\top$ and $\boldsymbol{\gamma}(k) = [\boldsymbol{\gamma}_1(k),\ \ldots,\ \boldsymbol{\gamma}_n(k)]^\top$. We can rewrite the system in (7.1) with the consensus protocol in (7.2) in a compact matrix form as

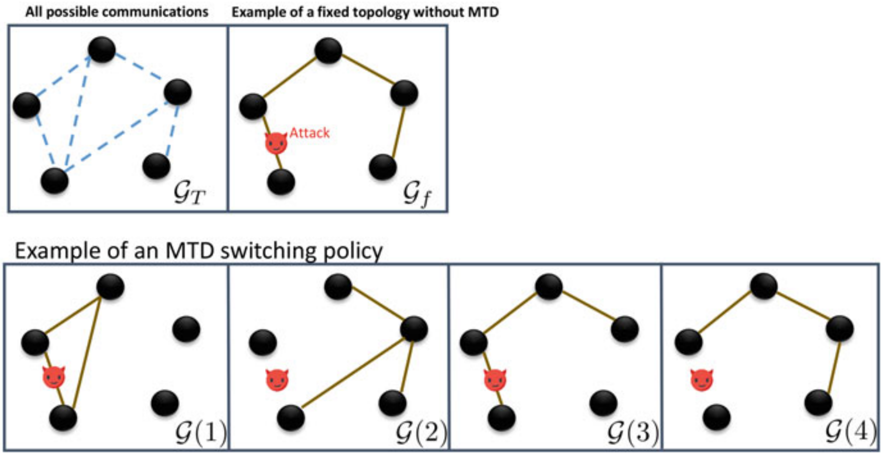

All possible communications
Example of a fixed topology without MTD
Attack
$\mathcal{G}_T$
$\mathcal{G}_f$
Example of an MTD switching policy
$\mathcal{G}(1)$
$\mathcal{G}(2)$
$\mathcal{G}(3)$
$\mathcal{G}(4)$

Fig. 7.3 The main idea behind the switching topology consists on changing the topology such that the number of times the compromised information is used decreases while guaranteeing stability of the system for the attack-free scenario. In this example only 50% of the times the fake compromised information can be transmitted. However, in the fixed case the attack is always affecting the communication between two nodes

$$z(k+1) = \underbrace{\begin{bmatrix} I & I \\ -\alpha_1 L(k) & I - \alpha_2 L(k) \end{bmatrix}}_{F(k)} z(k) + \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \alpha_1 I_n & \alpha_2 I_n \end{bmatrix}}_{G} \boldsymbol{\phi}(k),$$

$$z(k+1) = F(k)z(k) + G\boldsymbol{\phi}(k), \tag{7.3}$$

for $\boldsymbol{\phi}(k) = [\boldsymbol{\delta}(k)^\top, \boldsymbol{\gamma}(k)^\top]^\top$.

The main idea of MTD in multi agent systems is summarized in Fig. 7.3, where a communication graph with an MTD switching policy can mitigate the impact of an attack in the communication link by minimizing the amount of time the fake information is transmitted.

7.2.2 Random Graphs

A random graph $\mathscr{G}(k)$ is a graph generated by some random process [13]. Typically, the set of vertices $\mathscr{V}$ is assumed constant throughout time whereas the set of edges $\mathscr{E}(k)$ varies randomly with time. A general way of modeling the randomness of the edges consists in assuming a probability of connection between two vertices i and j, such that $a_{ij}(k) = 1$ is a Bernoulli random variable with probability $0 \leq p_{ij} \leq 1$. We can define the connection probability matrix $\boldsymbol{P} \in \mathbb{R}^{n\times n}$ with entries

$$\boldsymbol{P}_{ij} = \begin{cases} p_{ij}, & i \neq j \\ 0, & i = j. \end{cases}$$

Then, a realization $\mathscr{G}(k)$ at time k can be seen as a spanning subgraph (not necessarily connected) of the super graph $\mathscr{G}_T$. Due to the random nature of $\mathscr{A}(k)$, the instantaneous Laplacian matrix $L(k)$ is also random. The expected value of the adjacency matrix $E[\mathscr{A}(k)] = \boldsymbol{P}$ and the expected Laplacian matrix is then $\bar{L} = diag(\boldsymbol{P}\mathbf{1}_n) - \boldsymbol{P}$.

Erdös-Rényi Model

Erdös and Rényi [4] introduced two models of random graphs that consider two different ways of modeling the randomness of the edges:

1. The model $\mathscr{G}(k) = (\mathscr{V}, s)$ refers to a random graph with a fixed vertex set $\mathscr{V}$, where at each realization there exists exactly s edges. In other words, at each time k a graph $\mathscr{G}(k)$ is chosen uniformly at random from the collection of graphs that have n vertices and s edges.
2. The model $\mathscr{G}(k) = (\mathscr{V}, p)$ refers to a graph with vertex set $\mathscr{V}$ where each edge exists with nonzero probability p, equal for all vertices, such that for all i, j, $p_{ij} = p$.

We focus on a special case of the second Erdös-Rényi model, where only the edges that belong to $\mathscr{E}_T$ have probability p. In other words, $E[\mathscr{A}(k)] = p\mathscr{A}_T$ and $E[L(k)] = \bar{L} = p\mathscr{L}_T$. We refer to these types of graphs as MER (Modified Erdös-Rényi) graphs.

7.2.3 Convergence of the Attack-Free Scenario

It is necessary to guarantee that the inclusion of the proposed random MTD strategy does not affect the convergence to a consensus state. First, as it was pointed out in [25], convergence to a consensus state of a second-order model depends on the correct selection of α_1, α_2 and the connectivity properties of the communication topology, according to the following theorem adapted from [25] for fixed communication graphs.

Theorem 7.1 (Collorary 1 [25]) *Consider the multi-agent system in* (7.3) *without attack and with an undirected and fixed communication topology. Consensus can be achieved if and only if* $\alpha_2 > \alpha_1 > 0$ *and* $\alpha_1 - 2\alpha_2 > \frac{-4}{\mu_i}$ *for all i.*

Now, the following theorem extends Theorem 7.1 and establishes sufficient conditions for convergence in expectation in the presence of random switching topologies.

Theorem 7.2 *Let* $\mathscr{G}_T = (\mathscr{V}, \mathscr{E}_T)$ *be the communication graph that describes all possible communications between n agents, and let* $\mathscr{A}_T$ *be its adjacency matrix with Laplacian matrix* $\mathscr{L}_T$. *Let* $\mu_1 = 0 < \mu_2 \leq \ldots \leq \mu_n$ *be the eigenvalues of* $\mathscr{L}_T$.

Suppose that each communication link exists with identical probability p such that $E[\mathscr{A}_T] = P = p\mathscr{A}_T$ *and* $\bar{L} = p\mathscr{L}_T$. *The consensus state* $z_c = [x_c^\top \ v_c^\top]^\top$, *for*

$$\begin{aligned} x_c &= \mathbf{1}_N \left(\frac{1}{N} \sum_{j=1}^{n} x_j(0) + \frac{k}{N} \sum_{j=1}^{n} v_j(0) \right), \\ v_c &= \mathbf{1}_N \frac{1}{N} \sum_{j=1}^{n} v_j(0) \end{aligned} \tag{7.4}$$

is reached in expectation if $\alpha_1 = \frac{p}{\mu_n}$ *and* $\alpha_2 = \frac{1+p}{\mu_n}$.

Proof Let $\bar{z}(k) = E[z(k)]$ denote the expected state vector, such that the dynamics in (7.3) without attack can be rewritten as

$$\bar{z}(k+1) = \bar{F}\bar{z}(k)$$

where

$$\bar{F} = \begin{bmatrix} I & I \\ -\alpha_1 \bar{L} & I - \alpha_2 \bar{L} \end{bmatrix}.$$

Recall that $\bar{L}$ is the Laplacian matrix of an undirected graph and that the consensus state is reached for fixed topologies if $\alpha_1 > \alpha_2 > 0$ and $\alpha_1 - 2\alpha_2 > \frac{-4}{\bar{\mu}_i}$ according to Theorem 7.1, where $\bar{\mu}_i$ is the ith eigenvalue of $\bar{L}$ for $i = 2, \ldots, n$. Since $\mathscr{L}_T$ is symmetric, we have that $\bar{\mu}_i = p\mu_i$. Thus, $\frac{p}{\mu_n} - 2\frac{1+p}{\mu_n} > \frac{-4}{p\mu_n} > \frac{-4}{p\mu_i}$. Multiplying by $p\mu_n$, we obtain $-p^2 - 2p + 4 > 0$ which is always true for $0 < p \leq 1$. □

Remark 7.1 Convergence in expectation means that the speed $v(k)$ will converge to a vicinity of v_c.

Corollary 7.1 *When the random graph is described by an Erdös-Rényi model with degree s, then the states* $z(k)$ *will converge surely to* z_c, *i.e.,* $\Pr\{\lim_{k\to\infty} z(k) = z_c\} = 1$.

We have shown convergence conditions in expectation that depend on the correct selection of α_1, α_2. However, convergence in expectation is not enough to guarantee asymptotic behavior to a consensus state. Therefore, we introduce the following definition.

Definition 7.1 (Mean Square Consensus) Under random switching topologies, the multi-agent system in (7.1) reaches mean square consensus if, for any $i \neq j$, $|x_i(k) - x_j(k)| \to 0$ and $|v_i(k) - v_j(k)| \to 0$ hold in mean square sense for any initial states, such that the consensus state belongs to the vicinity of z_c.

The notion of mean square consensus ensures that $z(k)$ will converge asymptotically to a consensus state with probability 1, and the consensus state is in the vicinity of z_c.

To find conditions for mean square consensus, we will use the results stated in the following Theorem adapted from [26] for Markovian switching topologies.

Theorem 7.3 (Theorem 4 in [26]) *Assume the switching topology is driven by an ergodic Markov process (or a Bernoulli process). There exists gains* α_1, α_2, *such that under the linear protocol in* (7.2) *the multi-agent system in* (7.1) *reaches mean square consensus, if and only if the union of the graphs in the topology set of size r,* $\{G_1, G_2, \ldots, G_r\}$ *has a globally reachable node.*

Since our edge set is random and changes at each time instant k, we do not have a fixed set of communication topologies; however, if we can show that after a finite number of switches, the union of any random graph realizations has a globally reachable node, we can ensure mean square consensus.

Lemma 7.1 *For any MER (and Erdös-Rényi) graph* $\mathscr{G}(k) = (n, p)$*with* $p > 0$*, there exists a* $k^* < \infty$ *such that the union of graph realizations* $\mathfrak{G} = \mathscr{G}(1) \cup \mathscr{G}(2) \cup \ldots \cup \mathscr{G}(k^*)$*is connected.*

Proof Let $\mathfrak{E} = \{\mathscr{E}(1) \cup \mathscr{E}(2) \cup \ldots \cup \mathscr{E}(k^*)\}$ be the union of the edge sets with elements $\mathfrak{e}_{ij}$. Therefore, $\mathfrak{e}_{ij} \neq \emptyset$ if, for $k = 1, \ldots, k^*$, the link $e_{ij}(k)$ has existed at least once. It is easy to see that the union of modified Erdös-Rényi graphs $\mathfrak{G}$ is also a modified Erdös-Rényi random graph with the same vertex set and probability $\widetilde{p} = \Pr[\mathfrak{e}_{ij} \neq \emptyset]$. Since the existence of the edge $e_{ij}(k) \in \mathscr{E}(k)$ at an instant k is described by a Bernoulli random variable with probability p, then the probability that the link has existed at least once after k^* realizations is described by the complement of a binomial distribution, as follows

$$\begin{aligned} \Pr[\mathfrak{e}_{ij} \neq \emptyset] &= 1 - \Pr[(e_{ij}(k) \neq \emptyset) \leq 1, k^*] \\ &= 1 - (1-p)^{k^*} - k^* p (1-p)^{k^*-1}. \end{aligned} \tag{7.5}$$

where $\Pr[(e_{ij}(k) \neq \emptyset) \leq 1, k^*]$ is the probability that e_{ij} existed at most once after k^* trials.

Notice that

$$\lim_{k^* \to \infty} 1 - (1-p)^{k^*-1} \left(1 - p + k^* p\right) = 1 \tag{7.6}$$

such that $\mathfrak{G} \to \mathscr{G}_T$. However, we need to show that there exists a finite $k^* < \infty$ such that $\mathfrak{G}$ is connected with high probability. To this end, recall that for a typical Erdös-Rényi graph $G(k) = (n, p)$, there exists a threshold $p > \frac{\log n + c}{n} < 1$ such that $\Pr[G(k) = \mathit{connected}] \to e^{-e^{-c}}$ [4]. The proof is based on defining a random variable X_0 that counts the number of isolated vertices when all communications are possible, and finding the probability that $P[X_0 = 0]$. Therefore, for the modified Erdös-Rényi random graph $\mathfrak{G} = (n, \widetilde{p})$, we can apply the same methodology by

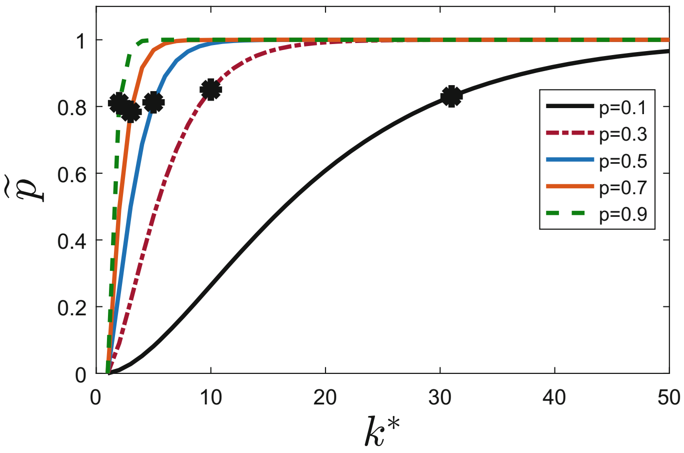


Fig. 7.4 Relationship between the instant k^* and the probability of an edge being connected after k^* iterations $\widetilde{p}$. When k^* increases, the union of random graphs (where each graph may be disconnected) forms a connected graph

restricting the edge set to $\mathscr{E}_T$, and find the specific threshold for which $\Pr[\mathfrak{G} = connected]$ with high probability. The calculation of that threshold is not an easy task, but since (7.6) holds, we know it exists. □

Example
Consider the modified ER random graph with $n = 10$, where $\mathscr{G}_T$ is a 2-regular graph. Figure 7.4 illustrates the relationship between k^* and $\widetilde{p}$ for different p. Clearly, as k^* increases, so does the probability of connection. Also, we calculated the number of iterations k^* until $\mathfrak{G}$ is connected and we repeated this process 1000 times. The asterisk indicates the maximum k^* associated to each probability. Clearly, for each p there is a finite k^* that ensures that $\mathfrak{G}$ is connected.

Now, we are able to state the following theorem.

Theorem 7.4 *The system in* (7.1) *with the consensus algorithm in* (7.2) *and with MTD policy described by the modified Erdös-Rényi graph with* $0<p<1$ *and total graph* $\mathscr{G}_T$ *reaches mean square consensus if* α_1, α_2 *are selected according to Theorem* 7.2*, and if* $\mathscr{G}_T$ *is connected.*

Proof Invoking Theorem 7.3, and since the union of modified Erdös-Rényi graphs is connected after a finite number of iterations for $p > 0$ according to Lemma 7.1, then there exists gains α_1, α_2 such that the second-order consensus algorithm is mean square stable and converge to a vicinity of z_c. From Theorem 7.2, we have found α_1, α_2 that guarantee stability in expectation. Thus, they also are sufficient to ensure mean square consensus. □

7.2.3.1 Convergence Rate with MTD

Using the proposed MTD strategy induces a deterioration of the convergence rate to the consensus state. Therefore, we will use the convergence rate as a measure of performance in order to identify how p and $\mathscr{G}_T$ affect the performance of the system.

Definition 7.2 The **convergence rate** in a consensus algorithm is the rate of convergence to the steady state value and it can be characterized by the spectral gap $R = 1 - \rho(F)$, where $\rho(F) = \max(|\lambda_i| : \lambda_i \neq 1)$. A convergence rate of 1 is the fastest possible convergence and 0 implies that the dynamics are not evolving at all.

In order to quantify the convergence rate in the presence of random switching, we introduce the following lemma adapted from [25].

Lemma 7.2 *Let us consider the second-order consensus algorithm described in* (7.3) *for fixed communication topology such that $L(k) = L$ with eigenvalues μ_i and $F(k) = F$. The eigenvalues of F are given by*

$$\lambda_{i1,2} = \frac{-\alpha_2\mu_i \pm \sqrt{\alpha_2^2\mu_i^2 - 4\alpha_1\mu_i}}{2} + 1$$

Proof The proof can be found in [25]. □

The degradation caused by using MTD can be calculated by comparing two cases, consensus with a fixed topology described by $\mathscr{G}_f$, and with a random topology. For the fixed topology, we consider the special case where all possible communications are active, such that $\mathscr{G}_f = \mathscr{G}_T$, as follows.

Lemma 7.3 *Let $\mathscr{G}_T$ be the graph that represents all possible communications and let us consider the special case where all possible communication links exist, i.e., the fixed communication topology without MTD is given by $\mathscr{G}_f = \mathscr{G}_T$. Applying the algorithm in* (7.3) *with fixed topology (i.e., $p = 1$) the convergence rate is $R_f = 1 - \rho(F)$, for $\rho(F) = \sqrt{1 - \frac{\mu_2}{\mu_n}}$.*

Proof From Lemma 7.2, we have that for a fixed topology with Laplacian matrix L, the eigenvalues of F are given by

$$\lambda_{i1,2} = \frac{-\alpha_2\mu_i \pm \sqrt{\alpha_2^2\mu_i^2 - 4\alpha_1\mu_i}}{2} + 1.$$

When $p = 1$, from Theorem 7.2 we have that $\alpha_1 = \frac{2}{\mu_n}$, $\alpha_2 = \frac{1}{\mu_n}$, such that

$$\lambda_{i1,2} = \frac{-2\frac{\mu_i}{\mu_n} \pm \sqrt{4\frac{\mu_i^2}{\mu_n^2} - 4\frac{\mu_i}{\mu_n}}}{2} + 1 = -\frac{\mu_i}{\mu_n} \pm \sqrt{\frac{\mu_i}{\mu_n}\left(\frac{\mu_i}{\mu_n} - 1\right)} + 1.$$

Notice that the term inside the square root is always negative, such that the eigenvalues have a component in the imaginary axis. We then can rewrite the eigenvalues as

$$\lambda_{i1,2} = -\frac{\mu_i}{\mu_n} + 1 \pm \mathbf{j}\sqrt{\frac{\mu_i}{\mu_n}\left(1 - \frac{\mu_i}{\mu_n}\right)}.$$

The magnitude is then

$$|\lambda_i| = \sqrt{\left(1 - \frac{\mu_i}{\mu_n}\right)^2 + \frac{\mu_i}{\mu_n}\left(1 - \frac{\mu_i}{\mu_n}\right)} = \sqrt{1 - \frac{\mu_i}{\mu_n}}.$$

Since all eigenvalues μ_i are real, $\rho(F) = \sqrt{1 - \frac{\mu_2}{\mu_n}}$ and $R_f = 1 - \rho(F)$. □

Now, the upper bound of the expected convergence rate is derived in the following theorem.

Theorem 7.5 *Suppose that an MTD random mechanism is introduced such that the communication topology changes randomly over time with probability p. Therefore, the expected convergence rate* $\bar{R}_{MTD} < R_f$ *for any* $0 < p < 1$ *is given by*

$$\bar{R}_{MTD} = 1 - \sqrt{1 - p\frac{\mu_2}{\mu_n}}.$$

Proof Since $\rho(F)$ is a convex function for nonnegative matrices, from the Jensen's inequality, we have that

$$E[\rho(F(k))] \geq \rho(E[F(k)]) = \rho(\bar{F}).$$

Therefore, $\bar{R}_{MTD} = E[R_{MTD}(k)] = 1 - E[\rho(F(k))] \leq 1 - \rho(\bar{F})$.

Now, suppose $0 < p < 1$, such that the eigenvalues of $\bar{F}$ are given by

$$\bar{\lambda}_{i1,2} = \frac{-(1+p)\frac{\bar{\mu}_i}{\mu_n} \pm \sqrt{(1+p)^2\frac{\bar{\mu}_i^2}{\mu_n^2} - 4p\frac{\bar{\mu}_i}{\mu_n}}}{2} + 1,$$

where $\bar{\mu}_i$ are the eigenvalues of $\bar{L}$. Since $\mathscr{L}_T$ is symmetric, then $\bar{\mu}_i = p\mu_i$. Following the same steps as before, it is easy to see that $|\bar{\lambda}_i| = \sqrt{1 - p\frac{\mu_i}{\mu_n}}$ and $\bar{R}_{MTD} \leq 1 - \sqrt{1 - p\frac{\mu_2}{\mu_n}}$. Clearly, $\bar{R}_{MTD} < R_f$ since

$$\sqrt{1 - p\frac{\mu_2}{\mu_n}} > \sqrt{1 - \frac{\mu_2}{\mu_n}}$$

holds for any $p < 1$, and there is a degradation in the convergence rate. □

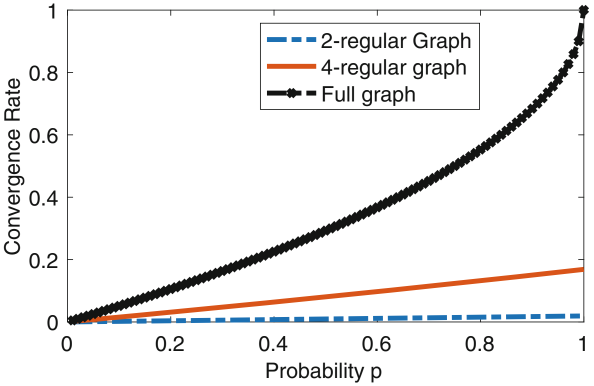


Fig. 7.5 Convergence rate for different probabilities and for several $\mathcal{G}_T$. Notice that increasing the communication capabilities in the network improves the consensus performance

Remark 7.2 Using MTD comes with a degradation in the convergence rate. Applications that require fast convergence to a consensus or a formation will need to select an appropriate large enough p.

Figure 7.5 shows the convergence rate for different graphs. Notice that the convergence rate increases with the connectivity of the communication graph, and decreases with p. As we will see next, p not only affects the convergence rate, but also the impact caused by an attacker. As a consequence, the defender needs to select appropriate p and $\mathcal{G}_T$ to maintain a good performance while making the system resilient to attacks.

7.2.4 Attack Impact with Random Switching Topology

We have calculated the convergence rate of the multi-vehicle system with a random switching communication topology in terms of the probability p. Clearly, to increase the convergence rate, it is necessary to select a large p. However, we need to quantify how the effect of a cyber-attack is affected by p in order to obtain a trade-off between the performance (convergence rate) and the impact of the attack.

Let x_c be the desired state or operational point at which the control action drives the system states. The main objective of an adversary is to deviate the system states from x_c. For instance, an adversary may intent to cause an increase on the pressure in a chemical reactor or cause that two vehicles crash. Therefore, we can define $\mathcal{I} \in \mathbb{R}_+$ as the impact that an attack can cause to the system as a function of $x(k) - x_c$. In this chapter, we define the impact as

$$\mathcal{I} = \lim_{k\to\infty} \|x(k) - x_c\|, \tag{7.7}$$

which captures effects of the attack even when stability is not compromised.

Now, suppose that the communication network in the multi-vehicle system changes randomly according to the model in (7.3). Let $E[z(k)] = \bar{z}$, $E[L(k)] = \bar{L}$, and $E[F(k)] = \bar{F}$, where

$$\bar{F} = \begin{bmatrix} I & I \\ -\alpha_1 \bar{L} & I - \alpha_2 \bar{L} \end{bmatrix}.$$

Similarly, we can define $E[\boldsymbol{\phi}(k)] = [E[\boldsymbol{\delta}(k)]^\top, E[\boldsymbol{\gamma}(k)]^\top] = \bar{\boldsymbol{\phi}}$ where

$$E[\boldsymbol{\delta}_i(k)] = p \sum\nolimits_{j=1}^{n} a_{ij}\alpha_1 E[\delta_{ij}(k)]$$

and

$$E[\boldsymbol{\gamma}_i(k)] = p \sum\nolimits_{j=1}^{n} a_{ij}\alpha_2 E[\gamma_{ij}(k)]$$

such that

$$\bar{z}(k+1) = \bar{F}\bar{z} + pG\boldsymbol{\phi}(k). \tag{7.8}$$

Theorem 7.6 *Let $\mathscr{G}_T = (\mathscr{V}, \mathscr{E}_T)$. be the communication graph that describes all possible communications between n agents, with Laplacian matrix $\mathscr{L}_T$. Let $\mu_1 = 0 < \mu_2 \leq \ldots \leq \mu_n$ be the eigenvalues of $\mathscr{L}_T$. Consider the system in (7.3) with a random topology with link connection probability $0 < p < 1$ and gains α_1, α_2 selected according to Theorem 7.2. The impact of the attack is given by*

$$\bar{\mathscr{I}} = \frac{p}{\mu_n}\sqrt{n(2p^2 + 2p + 1)}.$$

Proof The solution of (7.8) in the presence of an attack is given by

$$\bar{z}(k+1) = \bar{F}^k \bar{z}(0) + \sum_{l=0}^{k-1} \bar{F}^{k-l-1} pG\bar{\boldsymbol{\phi}}(k), \tag{7.9}$$

In [25] it has been shown that, if α_1, α_2 are properly selected,

$$\lim_{k\to\infty} \bar{F}^k = \begin{bmatrix} \mathbf{1}_n\xi^\top & \mathbf{1}_n\xi^\top k \\ \mathbf{0} & \mathbf{1}_n\xi^\top \end{bmatrix}, \tag{7.10}$$

where $\xi = \frac{\mathbf{1}_n}{\sqrt{n}}$ is the unique nonnegative left eigenvector of $\bar{L}$ associated with the eigenvalue 0. In order to quantify the impact of an attack, we focus our attention on how any attack can affect the vehicles speed. Thus, from (7.3), (7.9) and (7.10) we have that

$$\lim_{k\to\infty} \bar{F}^{k-l-1} pG = p\begin{bmatrix} \mathbf{1}_n\xi^\top & \mathbf{1}_n\xi^\top k \\ \mathbf{0} & \mathbf{1}_n\xi^\top \end{bmatrix}\begin{bmatrix} \mathbf{0} & \mathbf{0} \\ \alpha_1 I_n & \alpha_2 I_n \end{bmatrix} = p\begin{bmatrix} \alpha_1\mathbf{1}_n\xi^\top k & \alpha_2\mathbf{1}_n\xi^\top k \\ \alpha_1\mathbf{1}_n\xi^\top & \alpha_2\mathbf{1}_n\xi^\top \end{bmatrix}.$$

Taking only the part related to the vehicle speed for $G_2 = [\alpha_1 I, \alpha_2 I]$ and α_1, α_2 according to Theorem 7.2, the expected impact can be defined as

$$\bar{\mathscr{I}} = \|\mathbf{1}_n\xi^\top pG_2\| = \sqrt{np^2(\alpha_1^2+\alpha_2^2)} = \frac{p}{\mu_n}\sqrt{n(2p^2+2p+1)}.$$

□

Figure 7.6 shows the trade-off between the performance given by the convergence rate and the impact of the attack for different types of graphs. Notice that small probabilities will decrease the impact of the attack but at the cost of small convergence rates. On the other hand, the degree of connectivity of $\mathscr{G}_T$ has a significant impact on mitigating the effects of the attack. When $\mathscr{G}_T$ is a full graph, all communication links are possible, and the system is clearly more resilient than for any other topology. Thus, the system designer can increase the amount of possible communication channels in order to select smaller p that will not cause a significant performance degradation, while guaranteeing good resiliency to attacks. However,

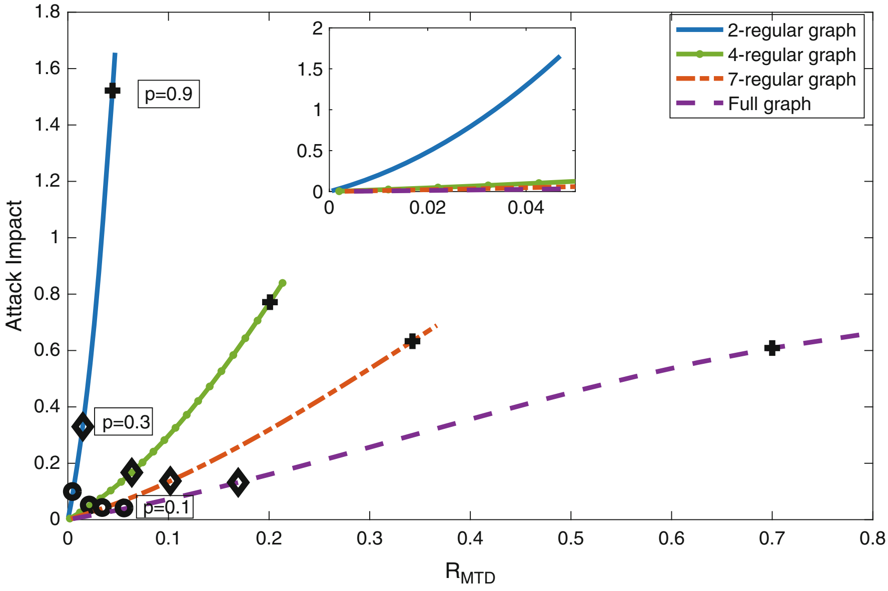


Fig. 7.6 Convergence rate vs. impact metric for different graphs with $n = 10$. Clearly, small p leads to lower vulnerability but at the cost of a decrease in the performance (decrease in the convergence rate). In particular, increasing the connectivity of the total graph, decreases considerably the impact of the attack

having a wide amount of communication channels for each vehicle may require more expensive equipment and more energy consumption.

7.3 Experiments

In order to illustrate the viability of our analysis, we consider two case studies, (1) vehicular platooning, and (2) UAVs formation control. In both scenarios, we show how the proposed random MTD strategy can be used to mitigate the impact of the attack.

7.3.1 Vehicular Platooning

We consider the problem of vehicular platooning. In particular, platooning offers many benefits over solo driving such as better reaction times, decrease of CO_2 emissions, and lower fuel consumption [18]. The objective of the platoon is to maintain an adequate distance between vehicles, such that sudden changes in the leader's speed (e.g., braking) will not cause any crash in the preceding vehicles. This is known as the string stability of the platoon and has been widely studied in the literature [11, 14, 19]. Typically, the Adaptive Cruise Control (ACC) system controls the distance and/or relative velocity between adjoining vehicles by measuring (radar/lidar) and reacting to the relative distance and/or velocity between adjacent vehicles compared to a desired setpoint. More recently, work has leveraged vehicle-to-vehicle or infrastructure-to-vehicle communication to inject feed-forward commands. Such Cooperative Adaptive Cruise Control (CACC) systems improve the string stability of the platoon and allow vehicles to follow each other with a closer distance than with ACC, thereby improving traffic flow capacity. CACC gathers information of vehicles further in front according to a specific communication network topology (Fig. 7.7).

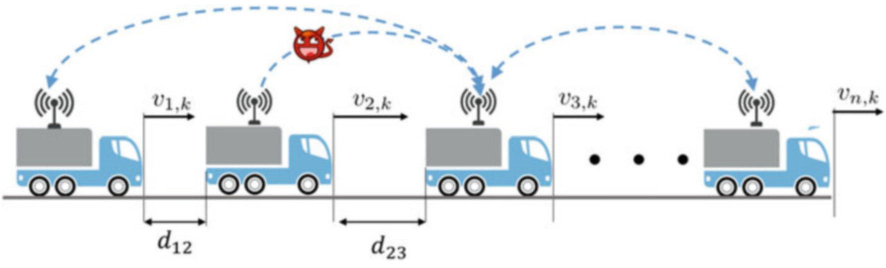


Fig. 7.7 Scheme of a platoon of n vehicles. Each vehicle is equipped with a CACC strategy using vehicle-to-vehicle communication network. An adversary can gain access to some sensors or actuator commands transmitted through the network

The dynamics of each vehicle in the platoon are dictated by (7.13) with a control strategy of the form

$$u_i(k) = -\alpha_1 \sum_{j=1}^{n} a_{ij}(k)\left(x_i(k) - x_j(k) - d_{ij}\right) - \alpha_2 \sum_{j=1}^{n} a_{ij}(k)\left(v_i(k) - v_j(k)\right),$$

where $d_{ij} = (j - i)d$ such that adjacent vehicles always have distance d [1].

In our simulations, we consider a platoon with 10 vehicles, $d = 2$m, and scenarios with and without MTD. Figure 7.8 illustrates the intra-vehicular distance $x_i - x_{i+1}$ and the speed of each vehicle. Before the attack, all intra-vehicle distances converge to $d = 2$ and to a speed of 72 km/h. Notice that including MTD affects the convergence time to the consensus state.

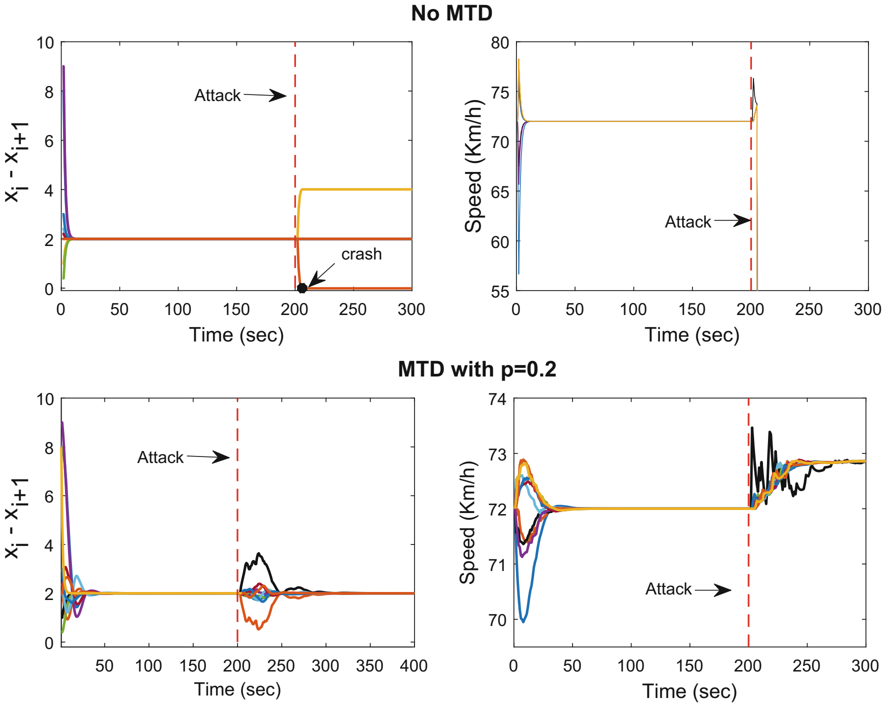


Fig. 7.8 Vehicles distance ($x_i - x_{i+1}$) and speed for the vehicular platooning problem, with desired $d = 2$ and final speed of 72 km/h. Vehicles reach the desired distance even with MTD, with a cost of slower convergence time. After 200 s an attacker compromises some of the communications received by vehicle 3 and launches a bias attack that fades over time. Without MTD, the attack causes that vehicles 2 and 3 crash (Top) causing the entire platoon to stop. On the other hand, with our proposed MTD and $p = 0.2$, the crash is avoided and the platoon speed slightly increases due to the attack (bottom)

Now, suppose that an adversary is able to compromise the information that agent 3 receives from one of its neighbors and injects a bias attack that fades over time. Without MTD, vehicles 2 and 3 crash after 5 s causing the entire platoon to stop (Fig. 7.8-Top). On the other hand, with a random MTD with $p = 0.2$, it is possible to avoid the crash and mitigate the impact of the attack. Notice that the attack only causes a slight increase in the speed and some oscillations but the consensus state is attained after the attack.

7.3.2 Formation Control of UAVs

Formations of UAVs have found use in military and civilian activities such as surveillance and exploration [8], building construction [24], and disaster management [15]. The main idea of these type of formations lies on the possibility that the group of UAVs moves as a single rigid body while performing a specific task using distributed and decentralized control strategies, where each UAV exchanges information only with a small group of agents.

To simply model the dynamics of n UAVs, we use (7.1) to represent the position and velocity in each axis, X and Y, respectively of each UAV [17], such that

$$\begin{aligned} x_{X,i}(k+1) &= x_{X,i}(k) + v_{X,i}(k) \\ v_{X,i}(k+1) &= v_{X,i}(k) + u_{X,i}(k) \\ x_{Y,i}(k+1) &= x_{Y,i}(k) + v_{Y,i}(k) \\ v_{Y,i}(k+1) &= v_{Y,i}(k) + u_{Y,i}(k), \end{aligned} \tag{7.11}$$

where $x_{X,i}(k)$, $v_{X,i}(k)$ are the position and speed in the X axis, $x_{Y,i}(k)$, $v_{Y,i}(k)$ are the position and speed in the Y axis.

For the formation control of UAVs, we assume that each UAV is able to control its speed in the X and Y axis separately, using a consensus-based algorithm of the form

$$\begin{aligned} u_{X,i}(k) = &- \alpha_1 \sum_{j=1}^{n} a_{ij}(k) \left(x_{X,i}(k) - x_{X,j} - d_{X,ij} \right) \\ &- \alpha_2 \sum_{j=1}^{n} a_{ij}(k) \left(v_{X,i}(k) - v_{X,j}(k) \right) \\ u_{Y,i}(k) = &- \alpha_1 \sum_{j=1}^{n} a_{ij}(k) \left(x_{Y,i}(k) - x_{Y,j} - d_{Y,ij} \right) \\ &- \alpha_2 \sum_{j=1}^{n} a_{ij}(k) \left(v_{Y,i}(k) - v_{Y,j}(k) \right) \end{aligned} \tag{7.12}$$

where $d_{X,ij}$, $d_{Y,ij}$ are the desired distances between each pair of agents that describe the desired formation. Since $d_{X,ij}$, dY, ij are fixed and since we assume that the states in each direction are independent, the stability analysis does not depend on the desired formation, but only on the selection of α_1, α_2 and p.

As an example, suppose we have 10 UAVs, each one with X,Y speed controls and the desired formation is a diamond shape at a height of 5 m. Each UAV possesses communication capabilities to transmit their XY position and both speeds in a single package where $\mathscr{G}_T$ is a 4-regular graph. Figure 7.9 depicts the $X - Y$ trajectories of the group of UAVs with the proposed MTD with $p = 0.2$ and without attack. Clearly, even in the presence of switching topologies, the desired formation is achieved.

Now, we assume an adversary compromises only the communication links that agent 3 receives from 1 after 200 s, with $\phi_{X,31} = 0.3$, $\gamma_{X,31} = 0.2$ for the X axis and $\phi_{Y,31} = -0.3\gamma_{Y,31} = -0.2$. The attack causes that the formation changes its direction by increasing the speed, as depicted in Figs. 7.10 and 7.11. However, the deviation can be mitigated for small p, at the cost of larger convergence times.

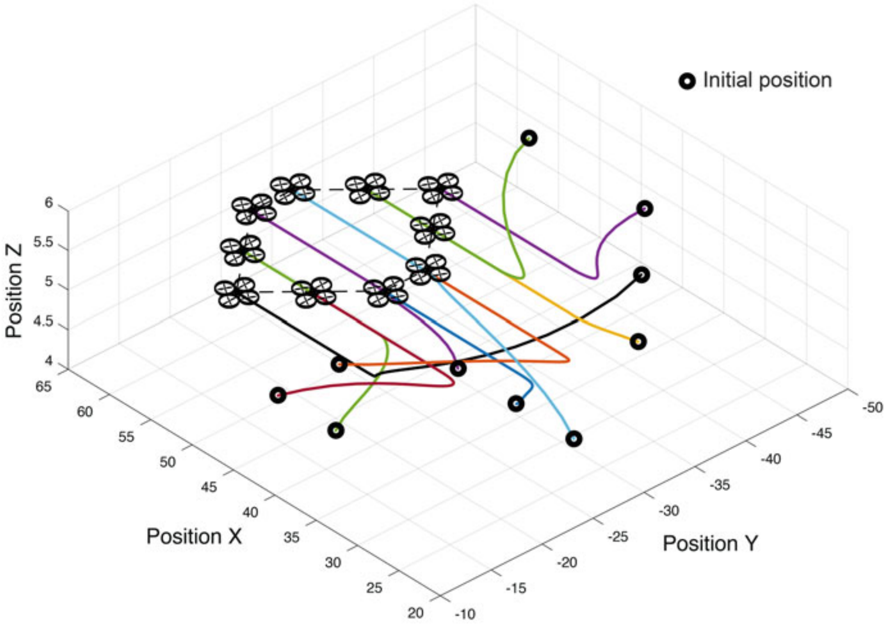


Fig. 7.9 Formation control of 10 UAVs that intend to form a diamond shape formation

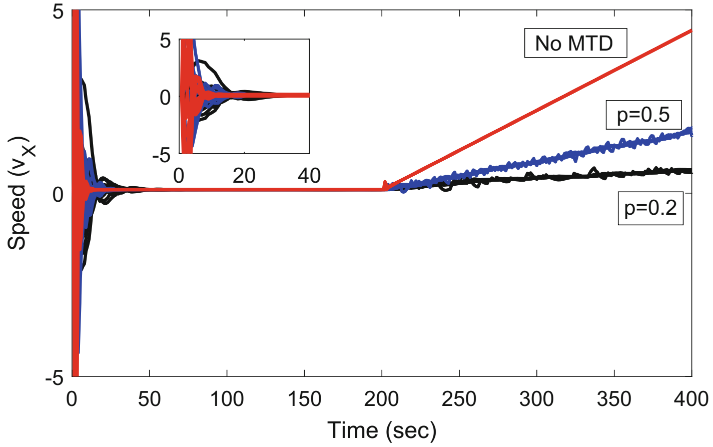


Fig. 7.10 Speed in the *X* axis of the group of UAVs with and without MTD. Before the attack, the control action guarantees a consensus in the speed and the desired formation is attained. After 200 s, an adversary compromises the information that agent 3 receives from one if his neighbors

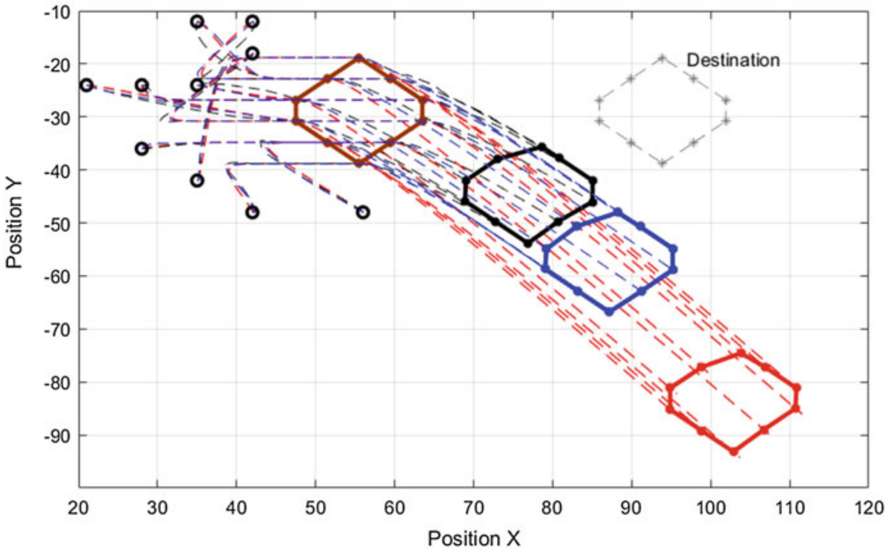


Fig. 7.11 Formation control of 10 UAVs with an MTD strategy for different $p = 0.2$ (black), $p = 0.5$ (blue), and no MTD (red) during 400 s. An attacker compromises one communication link received by agent 3 and launches a bias attack after 200 s. The group of agents is deviated from its destination at different speed rates depending on *p*. Clearly MTD decreases the impact caused by an attack that aims to deviate the formation

7.4 Toward Optimal Mitigation

We have shown how random switching of the communication topology in multi-agent systems can mitigate the deviation caused by cyber-attacks. Now, we want to extend the proposed strategy to a more general control systems frameworks where the switching can be performed at a sensor level (or in the communications between sensors and actuators), or at the controller level (e.g., performed by a PLC). Besides, we consider heterogeneous switching probabilities such that we can formulate optimization problems that allow us to find the *optimal probability distribution* of the switching strategy that decreases the impact caused by sensor attacks.

We consider a discrete-time linear time invariant (LTI) system described by

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) + \phi(k), \end{aligned} \tag{7.13}$$

where A, B, C are matrices of proper dimensions, and $x(k) \in \mathbb{R}^n$, $y(k) \in \mathbb{R}^p$, $u(k) \in \mathbb{R}^m$ are the state, output, and input vectors, respectively. Since the sensor/control commands can be sent through a communication network, we assume that the system can be subject to additive sensor attacks $\phi(k) \in \mathbb{R}^p$.

7.4.1 MTD in the Controller

We now consider the case where uncertainties are added to the system through the controller actions. The general architecture is illustrated in Fig. 7.12, where the control action is chosen from a group of appropriate controllers. Our objective is to design the sequence of control gains that can decrease the state deviation caused by sensor attacks. A similar MTD approach has been proposed in [6], where the authors design the random switching strategies that maximizes the entropy or unpredictability caused by the MTD mechanism.

Suppose we have n_c different control modes and let $\sigma(k) \in \mathbb{Z}_+$ for $\sigma(k) \leq n_c$ be the index of the control mode at the kth time instant. Let $\Sigma = \{\sigma(0), \sigma(1), \ldots\}$ denote the switching sequence or switching logic that orchestrate the different mode changes between controllers. Thus, we can define the control action as

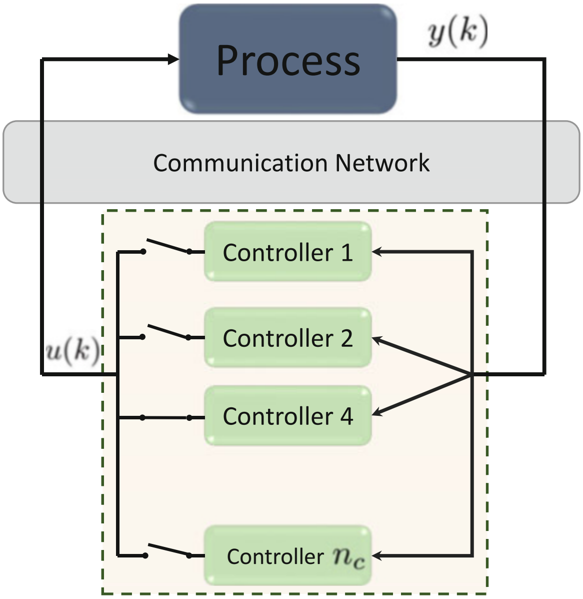
$$u(k) = K_{\sigma(k)} y(k), \tag{7.14}$$

where at each time instant k, the control gain is given by $K_{\sigma(k)} \in \mathfrak{K}$, for $\mathfrak{K} = \{K_1, K_2, \ldots, K_{n_c}\}$. Therefore, combining (7.13) and (7.14) we obtain

$$x(k+1) = (A + BK_{\sigma(k)}C)x(k) = F_{\sigma(k)}x(k) \tag{7.15}$$

where $F_{\sigma(k)} = A + BK_{\sigma(k)}C$.

Fig. 7.12 MTD scheme for switching among different controllers



The challenge with these type of linear systems lies on guaranteeing stability for an arbitrary index sequence Σ. Lin et al. [10] summarized some important results in the stability of the switched system in (7.15). The next theorem adapted from [9] states the necessary and sufficient condition for asymptotic stability.

Theorem 7.7 *The switched system in* (7.15) *is asymptotically stable under an arbitrary switching if and only if there exists an arbitrary integer* $\boldsymbol{n}$ *such that for all* $\boldsymbol{n}$*-tuple* $F_{i_j} \in \{F_1, F_2, \ldots, F_{n_c}\}$ *for* $j = 1, \ldots, \boldsymbol{n}$

$$\|F_{i_1} F_{i_2} \ldots F_{i_n}\| < 1.$$

The question now is, how can we limit the impact of cyber-attacks by switching among controllers?

To answer this question, we need to solve two problems: (1) find the set of controllers $\mathfrak{K}$, and (2) find the switching sequence. We propose an approach that solves both problems as a motivation to show how MTD can decrease the impact of attacks. To solve the first problem, and since we are trying to limit the impact of sensor attacks, we assume that the elements of an optimal control gain (e.g., LQR controller) can be active or inactive, such that not all sensor data is used at each time instance. For instance, if an optimal control (without switching) is $K_T = [K_{T1}, K_{T2}, K_{T3}]$, we can assume that $K_1 = [0, K_{T2}, K_{T3}]$, $K_2 = [K_{T1}, 0, K_{T3}]$, and $K_3 = [K_{T1}, K_{T2}, 0]$. In this way, we do not use all the sensor information at all times. Therefore, if conditions of Theorem 7.7 are satisfied for the control set, any arbitrary switching sequence guarantees asymptotic stability. If we consider observer-based controllers, the same technique can be applied to the estimation gain L.

For the second problem, we will consider random switching, such that we can exploit some tools from stochastic systems. Let p_j be the probability that control $K_j \in \mathfrak{K}$ is active where $\sum_{j=1}^{n_c} p_j = 1$. Suppose that $\bar{x}(k) = E[x(k)]$ denotes the expected state, such that

$$\bar{x}(k+1) = \bar{F}\bar{x}(k)$$

where

$$\bar{F} = E[F_{\sigma(k)}] = A + B\bar{K}C$$

and $\bar{K} = \sum_{j=1}^{n_c} p_j K_j$. Therefore, the design of the switching mechanism becomes the design of an appropriate probability distribution that assigns probabilities to each controller.

In the presence of a sensor attack, we have

$$\bar{x}(k+1) = \bar{F}x(k) + B\bar{K}E[\phi(k)],$$

with solution

$$\bar{x}(k) = \bar{F}^k\bar{x}(0) + \sum_{l=1}^{k-1} \bar{F}^{k-l-1}B\bar{K}E[\phi(l)]. \tag{7.16}$$

Assuming that $E[\phi(k)] = \bar{\phi}$ is constant for all k, and combining (7.7) with (7.16) for $x_c = 0$ and for $\bar{x}(k)$ we can calculate the expected impact

$$\bar{\mathscr{I}} = \lim_{k\to\infty} \|\bar{x}(k)\| \leq \|(I - \bar{F})B\bar{K}\|\|\bar{\phi}\|,$$

such that we can formulate the following nonlinear optimization problem:

Problem 1

$$\min_{p_1,p_2,\ldots,p_{n_c}} \quad \|(I - \bar{F})^{-1}B\bar{K}C\| \qquad s.t. \tag{7.17}$$

$$\sum_{i=1}^{n_c} p_i = 1, \qquad p_i \geq 0, \quad \text{for all} \quad i. \tag{7.18}$$

The solution of Problem 1 provides the probability distribution of the random switching strategy that minimizes the effects of an adversary in expectation. This formulation can be extended to include performance constraints or additional objectives, such as the entropy metric proposed in [6].

Remark 7.3 When $E[\phi(k)]$ is not constant, we can consider other impact metrics such as the sensitivity of the H_∞ gain.

Example 1

Consider the linear system described by

$$A = \begin{bmatrix} 0.7 & -0.5 & 0 \\ 0.2 & 0.8 & 0.3 \\ 0.4 & 0.2 & 0.7 \end{bmatrix}, \quad B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, C = I, \quad K_T = -[0.14, 0.17, 0.16]$$

where K_T is an LQR control. Suppose that an adversary gains access to sensors 1 and 2 and injects a sensor bias attack $\phi = [1, 1, 0]^\top$. We assume that the system possesses an MTD strategy that selects between a set of controllers $K_1 = -[0,\ 0.17,\ 0.6]$, $K_2 = -[0.14,\ 0,\ 0.16]$, $K_3 = -[0.14,\ 0.17,\ 0]$. In total we have 3 possible control gains that are randomly selected at each time instant. Solving Problem 1, we found the switching probability distribution $p = [0.91,\ 0.04,\ 0.05]$. Figure 7.13 illustrates how switching among controllers can help to mitigate the effects of the attack by decreasing the deviation caused by the adversary. We use $\|x(k)\|$ to measure the total state deviation at each time instant. Notice that MTD comes with a performance cost by decreasing the convergence rate to the equilibrium, but it decreases the total state deviation.

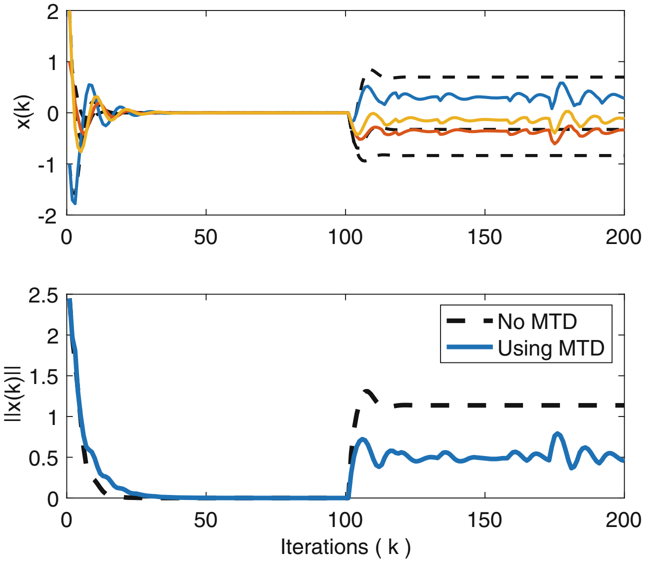


Fig. 7.13 States and energy of a linear system with an MTD that switches among three different control strategies (solid line), and with a fixed LQR control (dashed). Clearly, MTD decreases the deviation caused by the adversary at the cost of performance degradation

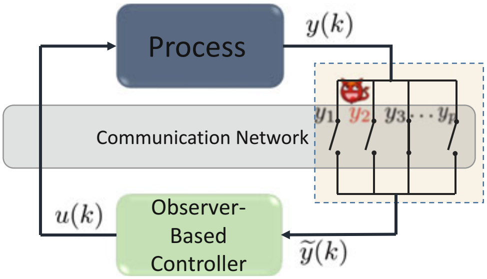


Fig. 7.14 MTD scheme for sensor switching. Each sensor can be connected (transmit data at a specific time instant) or disconnected (not transmitting)

7.4.2 MTD in Sensors

In this case, the MTD mechanism can arbitrarily break the communication between a subset of sensors and the controller at any time k, as depicted in Fig. 7.14. For instance, suppose sensor y_2 is compromised. If the probability that the communication link between y_2 and the controller exists is low, then the amount of fake information received by the controller (or estimator) will decrease and the effects of that attack in the control command may be mitigated.

Suppose we have m sensors and the communication link between any sensor and the controller may be active or inactive. Let $\mathscr{C} = \{C_1, C_2, \ldots, C_{n_s}\}$ be the desired set of matrices that combine active or inactive sensors. Let $\theta(k) \in \mathbb{Z}_+$ be the index of the output modes at the kth time instant. Let $\Theta = \{\theta(0), \theta(1), \ldots\}$ denote the switching logic that changes among different sensors subsets. The output is then given by $\widetilde{y}(k) = C_{\theta(k)}x(k)$, where $C_{\theta(k)} \in \mathscr{C}$. Therefore, the linear system in (7.13) becomes

$$x(k+1) = (A + BKC_{\theta(k)})x(k) = G_{\theta(k)}x(k). \tag{7.19}$$

Notice that (7.15) and (7.19) are similar, and the asymptotic stability of (7.19) can be guaranteed if conditions in Theorem 7.7 are satisfied for $G_{\theta(k)}$.

Similar to the case with switching actuators, we will assume that each sensor is active with probability p_i, for $i = 1, \ldots, m$, such that we can define the matrix $\boldsymbol{P} = diag(p_1, p_2, \ldots, p_m)$. To facilitate the analysis, we can define $\mathscr{S}(k)$ as the diagonal matrix with elements $s_{ii}(k) = 1$ if sensor i is active at the instant k, and 0 otherwise. When the system is subject to a sensor attack $\phi(k)$, we can rewrite (7.19) as

$$x(k+1) = (A + BKS(k)C)x(k) + BKS(k)\phi(k),$$

where C is the output matrix without MTD and $S(k)C = C_{\theta(k)}$. Notice that $E[S(k)] = \boldsymbol{P}$, such that $\bar{C} = E[S(k)C] = \boldsymbol{P}C$ and $\bar{G} = A + BK\bar{C}$. The expected state dynamics are then given by

$$\bar{x}(k+1) = \bar{G}\bar{x}(k) + BK\boldsymbol{P}\bar{\phi}.$$

Now, we can formulate an optimization problem that aims to find $\boldsymbol{P}$ that minimizes the impact of the attack while preserving performance conditions $\mathscr{F}(A, B, C, K, \boldsymbol{P}) < \beta$ (e.g., expected spectral radius $\rho(\bar{G})$) as follows:

Problem 2

$$\begin{aligned} &\min_{p_1, p_2, \ldots, p_{n_s}} \|(I - \bar{G})^{-1} BK\boldsymbol{P}\| \\ &s.t. \\ &\mathscr{F}(A, B, C, K, \boldsymbol{P}) < \beta \\ &0 \leq p_i \leq 1, \quad \text{for all} \quad i. \end{aligned} \tag{7.20}$$

Example 2

Suppose A, B are the same from example 1, but now the system has 4 sensors, with output matrix and control gain given by

$$C \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 1 & 1 \end{bmatrix}, \quad K = -[0.14, 0.17, 0.16, 0.13].$$

Solving Problem 2 for $\mathscr{F} = \rho(\bar{G})$ and $\beta = 0.92$ we obtain $p = [0.06, 0.11, 0.07, 0.2]$. Figure 7.14 illustrates how MTD strategies can decrease the impact caused

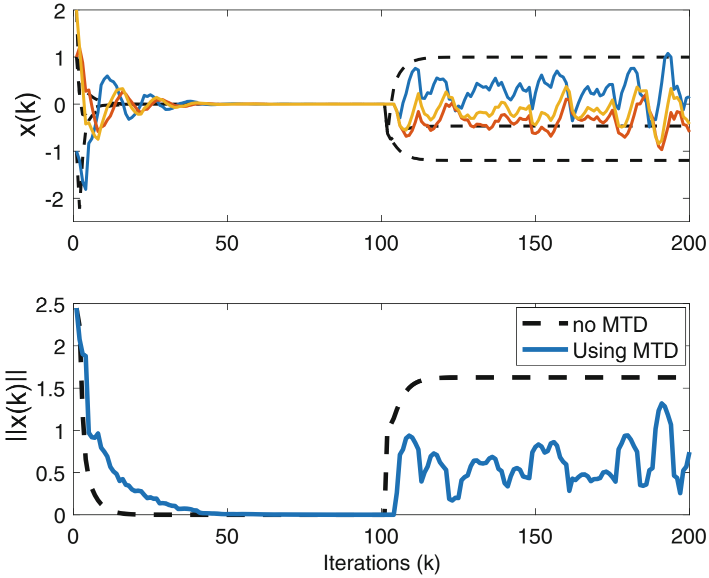


Fig. 7.15 States and energy of a linear system with an MTD with sensor switching and $p = [0.06, 0.11, 0.07, 0.2]$ (solid line), and without MTD (dashed lines). MTD mitigates the deviation caused by the adversary

by an adversary that injects an attack in all sensors $\phi = [1, 1, 1, 1]^\top$. Dashed lines correspond to the case without MTD (Fig. 7.15).

7.5 Conclusions and Future Directions

In this chapter, we have proposed an MTD strategy that randomly switches between different communication topologies in order to mitigate the deviation caused by an adversary. We have identified the trade-off between MTD and the convergence rate such that a system designer can choose adequate parameters that maintain specific levels of performance. In particular, from our analysis we found out that high connectivity of the graph $\mathscr{G}_T$ describing all possible communications and the low probability p play an important role in making the system more resilient to cyber-attacks with good convergence rate. We have also introduced two MTD strategies for more general feedback-control systems and we have proposed optimization problems that allow us to find the optimal probability distribution for the random switching mechanism.

There are many research directions that can be derived from the work presented in this chapter. In future work, we will consider heterogeneous probabilities for the multi-vehicle problem and find a relationship between the topology $\mathscr{G}_T$ and the matrix $\boldsymbol{P}$. Besides, we will consider more realistic models of multi-vehicle systems that include collision avoidance control, actuator saturation, and more complex dynamics. Finally, we will study how our proposed random MTD can affect anomaly detection mechanisms and design detection strategies that can leverage the use of MTD for CPS.

Acknowledgement This work was supported by the Air Force Office of Scientific Research under award number FA9550-17-1-0135.

References

1. S. Dadras, R.M. Gerdes, R. Sharma, Vehicular platooning in an adversarial environment, in *Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security* (ACM, New York, 2015), pp. 167–178
2. K.R. Davis, K.L. Morrow, R. Bobba, E. Heine, Power flow cyber attacks and perturbation-based defense, in *Proceedings of the IEEE Third International Conference on Smart Grid Communications (SmartGridComm), 2012* (IEEE, Piscataway, 2012), pp. 342–347
3. M. Dunlop, S. Groat, W. Urbanski, R. Marchany, J. Tront, Mt6d: a moving target ipv6 defense, in *Proceedings of the Military Communications Conference, 2011-Milcom 2011* (IEEE, Piscataway, 2011), pp. 1321–1326
4. P. Erdös, A. Rényi, On random graphs, I. Publ. Math. Debr. **6**, 290–297 (1959)
5. S. Jajodia, A.K. Ghosh, V. Swarup, C. Wang, X.S. Wang, *Moving Target Defense: Creating Asymmetric Uncertainty for Cyber Threats*, vol. 54 (Springer, New York, 2011)

6. A. Kanellopoulos, K.G. Vamvoudakis, Entropy-based proactive and reactive cyber-physical security, in *Proactive and Dynamic Network Defense*, ed. by C. Wang, Z. Lu. Advances in Information Security, vol. 74 (Springer, Cham, 2019). https://doi.org/10.1007/978-3-030-10597-6_3
7. S. Kar, J.M.F. Moura, Sensor networks with random links: topology design for distributed consensus. IEEE Trans. Signal Process. **56**(7), 3315–3326 (2008)
8. T. Kopfstedt, M. Mukai, M. Fujita, C. Ament, Control of formations of UAVs for surveillance and reconnaissance missions. IFAC Proc. Vol. **41**(2), 5161–5166 (2008)
9. H. Lin, P.J. Antsaklis, Stability and persistent disturbance attenuation properties for a class of networked control systems: switched system approach. Int. J. Control **78**(18), 1447–1458 (2005)
10. H. Lin, P.J. Antsaklis, Stability and stabilizability of switched linear systems: a survey of recent results. IEEE Trans. Autom. control **54**(2), 308–322 (2009)
11. S. Öncü, J. Ploeg, N. van de Wouw, H. Nijmeijer, Cooperative adaptive cruise control: network-aware analysis of string stability. IEEE Trans. Intell. Transp. Syst. **15**(4), 1527–1537 (2014)
12. Z.-H. Pang, G.P. Liu, Z. Dong, Secure networked control systems under denial of service attacks. IFAC Proc. Vol. **44**(1), 8908–8913 (2011)
13. S.S. Pereira, A. Pagès-Zamora, Mean square convergence of consensus algorithms in random WSNs. IEEE Trans. Signal Process. **58**(5), 2866–2874 (2010)
14. J. Ploeg, D.P. Shukla, N. van de Wouw, H. Nijmeijer, Controller synthesis for string stability of vehicle platoons. IEEE Trans. Intell. Transp. Syst. **15**(2), 854–865 (2014)
15. M. Quaritsch, K. Kruggl, D. Wischounig-Strucl, S. Bhattacharya, M. Shah, B. Rinner, Networked UAVs as aerial sensor network for disaster management applications. e & i Elektrotechnik und Informationstechnik **127**(3), 56–63 (2010)
16. M.A. Rahman, E. Al-Shaer, R.B. Bobba, Moving target defense for hardening the security of the power system state estimation, in *Proceedings of the First ACM Workshop on Moving Target Defense* (ACM, New York, 2014), pp. 59–68
17. W. Ren, R.W. Beard, *Distributed Consensus in Multi-Vehicle Cooperative Control* (Springer, London, 2008)
18. C. Suthaputchakun, Z. Sun, M. Dianati, Applications of vehicular communications for reducing fuel consumption and CO_2 emission: the state of the art and research challenges. IEEE Commun. Mag. **50**(12), 108–115 (2012)
19. D. Swaroop, J.K. Hedrick, String stability of interconnected systems. IEEE Trans. Autom. Control **41**(3), 349–357 (1996)
20. J. Tian, R. Tan, X. Guan, T. Liu, Hidden moving target defense in smart grids, in *Proceedings of the 2nd Workshop on Cyber-Physical Security and Resilience in Smart Grids* (ACM, New York, 2017), pp. 21–26
21. J. Valente, A.A. Cárdenas, Using visual challenges to verify the integrity of security cameras, in *Proceedings of the 31st Annual Computer Security Applications Conference*, ACSAC 2015, New York (ACM, New York, 2015), pp. 141–150
22. H. Wang, Q. Jia, D. Fleck, W. Powell, F. Li, A. Stavrou, A moving target DDoS defense mechanism. Comput. Commun. **46**, 10–21 (2014)
23. S. Weerakkody, B. Sinopoli, Detecting integrity attacks on control systems using a moving target approach, in *Proceedings of the IEEE 54th Annual Conference on Decision and Control (CDC)* (IEEE, Piscataway, 2015), pp. 5820–5826
24. J. Willmann, F. Augugliaro, T. Cadalbert, R. D'Andrea, F. Gramazio, M. Kohler, Aerial robotic construction towards a new field of architectural research. Int. J. Archit. Comput. **10**(3), 439–459 (2012)
25. D. Xie, S. Wang, Consensus of second-order discrete-time multi-agent systems with fixed topology. J. Math. Anal. Appl. **387**(1), 8–16 (2012)
26. Y. Zhang, Y.-P. Tian, Consentability and protocol design of multi-agent systems with stochastic switching topology. Automatica **45**(5), 1195–1201 (2009)

Chapter 8
The Role of Machine Learning and Radio Reconfigurability in the Quest for Wireless Security

Francesco Restuccia, Salvatore D'Oro, Liyang Zhang, and Tommaso Melodia

Abstract Wireless networks require fast-acting, effective and efficient security mechanisms able to tackle unpredictable, dynamic, and stealthy attacks. In recent years, we have seen the steadfast rise of technologies based on machine learning and software-defined radios, which provide the necessary tools to address existing and future security threats without the need of direct human-in-the-loop intervention. On the other hand, these techniques have been so far used in an ad hoc fashion, without any tight interaction between the attack detection and mitigation phases. In this chapter, we propose and discuss a Learning-based Wireless Security (LeWiS) framework that provides a closed-loop approach to the problem of cross-layer wireless security. Along with discussing the LeWiS framework, we also survey recent advances in cross-layer wireless security.

8.1 Introduction

Due to the broadcast nature of radio-frequency (RF) waves, wireless networks are particularly vulnerable to a plethora of security threats, including jamming, denial-of-service (DoS), eavesdropping, message falsification/injection, and address spoofing, just to name a few [106, 110, 113, 115]. These threats, when carried out stealthily, may disrupt the network's functionality and seriously compromise users' security and privacy.

Traditionally, wireless attacks have focused on the disruption of a single layer on the network protocol stack by concentrating all the adversary efforts on a single objective. For example, most of the existing jamming techniques focus on disrupting wireless communications by transmitting high-power RF waves on the physical medium [84]. Recently, a number of *cross-layer* wireless attacks [104, 105, 109]

F. Restuccia · S. D'Oro · L. Zhang · T. Melodia (✉)
Institute for the Wireless Internet of Things, Northeastern University, Boston, MA, USA
e-mail: frestuc@northeastern.edu; s.doro@northeastern.edu; liyangzh@ece.neu.edu; melodia@northeastern.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_8

has been proposed, where *activities* and *objectives* entail different layers of the network protocol stack. The main feature of cross-layer attacks is that they are carried out by attacking layers different than the targeted one (also called *helping layers*). As a consequence, small-scale (and thus, hard-to-detect) attack activities may lead to dramatic changes on the *target layer*. Cross-layer threats are further exacerbated by the fact that many network protocols functionalities such as power allocation, channel selection, and routing decisions are jointly optimized with a common objective [23, 51, 64], resulting in layers that are closely coupled with each other. As long as the helping and target layers are coupled, the attack will lead to the defender's *responsive* change on the target layer. Thus, with carefully-tuned attack activities and objectives, the defender's reaction will favor the attacker's objective.

Cross-layer attacks present unique challenges that cannot be addressed by legacy security techniques. First, cross-layer attacks leverage small-scale activities in the helping layer to achieve significant damage in the target layer. This implies that the attacker can achieve the same goal with relatively small-scale activities, and therefore remain *undetected*. On the other hand, existing attack detection methods often assume that attacks are conducted always in the same manner and always have the same objective, and that large-scale attacks have to be conducted in order to achieve substantial results [110]. This is not necessarily true in cross-layer attacks. Therefore, developing *detection* and *mitigation* algorithms able to swiftly detect and counteract small-scale, dynamic cross-layer attack activities is now more important than ever.

The main issue with legacy security countermeasures is that they are usually tailored to address specific threats under specific network circumstances defined *a priori* [74]. On the other hand, the reality is that *malicious activities are usually extremely dynamic in nature and thus cannot be fully addressed beforehand.* As wireless attacks become ever more sophisticated, next-generation wireless networks will need to abandon generalized, one-size-fits-all, bolted-on security and optimization mechanisms, and rely on "smart", dynamic solutions able to harness the synergy between hardware and software reconfigurability to provide reliable, efficient and effective cross-platform and cross-layer security solutions. This aspect hinders significantly the integration and coordination of different wireless networking technologies to maximize network capacity, reliability, security, and coverage, and prevent the provision of a true networking-as-a-service vision. For this reason, software-defined radio techniques to simplify network control and to make it easier to introduce and deploy new applications and services, and machine learning to provide adaptability and fast-time reaction to adversarial action.

Recently, *machine learning techniques* have exhibited unprecedented success in classification problems in areas such as speech recognition [21], spam detection [18], computer vision [35], fraud detection [3], and computer networks [5], among others. One of the main reasons behind machine learning's popularity is that it provides a general framework to solve very complex classification problems where a model of the phenomenon being classified is too complex to derive or too dynamic to be summarized in mathematical terms [22, 46, 112]. Almost in parallel with machine learning's development, the development of algorithms and protocols based on *software-defined radios* [43, 114] has gained tremendous momentum in the networking research community over the last years [58]. A software-defined radio is a wireless communication system where components that have been typically implemented in hardware (e.g. mixers, filters, amplifiers, etc.) are implemented in software to ensure fast reconfigurability and adaptation to critical channel conditions (e.g., significant multipath, Doppler effect, or path loss). The main downside of pure software-based solutions is that they completely trade-off reconfigurability for efficiency. On the other hand, we have recently seen a tremendous rise of wireless platforms based on the *system-on-chip* (SoC) concept [59]. These SoC platforms allow the design and implementation of customized hardware on the field-programmable gate array (FPGA) portion of the platform to achieve better performance [63].

Although existing work has used machine learning and software-defined radios to design wireless security systems, these approaches have been used in an ad-hoc manner (i.e., to solve a specific wireless attack). On the contrary, future wireless networks will need to use context-aware, adaptive security measures able to sense the environment and swiftly respond to a range of dynamic, unpredictable, cross-layer attacks. We envision a radically different approach to the design of wireless security systems that can deploy various defense strategies, depending on the network's protection needs, and ability to tolerate and manage the specific technique's dynamic configuration.

In this chapter we address the lack of a unifying, systematic approach to cross-layer wireless security by proposing and discussing a *Learning-based Wireless Security* (LeWiS) framework. First, we provide background notions on the enabling technologies for LeWiS in Sect. 8.2. Then, we discuss a taxonomy of relevant existing wireless networks in Sect. 8.3. We then provide an overview of the LeWiS framework in Sect. 8.4, and delve deeper into the learning-based control module of LeWiS by discussing its detection (Sect. 8.5) and mitigation (Sect. 8.6) modules. We conclude the chapter in Sect. 8.7.

8.2 Background on Enabling Technologies for LeWiS

In this section, we provide a brief survey of the technologies that are at the basis of LeWiS, i.e., software-defined radios and networking (Sect. 8.2.1), system-on-chip technologies (Sect. 8.2.2), and machine learning (Sect. 8.2.3).

8.2.1 Software-Defined Radios and Networking

Software-defined radios are generally defined as devices where frequency band, air interface protocol and functionality can be upgraded with software updates instead of a complete hardware replacement. The peculiar characteristic of software-defined radios is that they are capable of being re-programmed or reconfigured to operate with different waveforms and protocols through dynamic loading of new waveforms and protocols. Furthermore, protocol characteristics at the data-link, network and transport are completely defined in software and may be changed dynamically at runtime depending on the system's needs. This has several advantages, including the ease of portability of the software to different hardware platforms, which in turn decreases development and testing costs as well as time-to-market of the radio technology. The software-defined radio technology uses modules that run on a generic hardware platform consisting of digital signal processing (DSP) processors as well as general purpose processors to implement the radio functions to transmit and receive signals. Usually, software-defined radio implementations are based on a combination of FPGA/DSP or FPGA-only solutions in alliance with software-defined physical and data-link layer for reaching the trade-off between parameter adaptation and performance. For an excellent and recent tutorial on the topic, the reader is referred to [99].

8.2.2 System-on-Chip (SoC) Technologies

System-on-chips (SoCs) [59] are embedded devices where a general-purpose processing system resides on the same integrated circuit with a field-programmable gate array (FPGA). SoCs are one of the leading technologies on the market for the implementation of digital systems combining software parts with hardware accelerators. The latest generations of these embedded devices include, among many other useful resources, powerful embedded hard processors supporting different operating systems, analog front-ends, specialized hardware blocks for high-performance computing or crypto-acceleration, and communication interfaces compatible with the most widely used network protocols.

The programmable logic implemented on the FPGA enables the efficient implementation of systems that perfectly fit the heterogeneous nature of wireless applications. This is because both hardware and software components can be configured according to the needs of different target applications; they are relatively low cost, low power, and compact, and their flexibility and possibility of code reuse (both hardware and software) allow the time to market to be reduced. Parallelism is another significant advantage of FPGAs. The distributed nature of the logic and interconnect resources in an FPGA fabric, together with the inherent concurrency of the hardware, allows several blocks operating in parallel (with either the same or different functionalities) to be implemented on a single chip.

8.2.3 Machine Learning

The pioneer of machine learning, Arthur Samuel, defined it as a "field of study that gives computers the ability to learn without being explicitly programmed"

[68]. Machine learning focuses on classification and prediction based on known properties, and usually consists of two phases: training and testing. Often, the following steps are performed: (1) identify class attributes (features) and classes from training data; (2) identify a subset of the attributes necessary for classification (i.e., dimensionality reduction); (3) learn the model using training data; and (4) use the trained model to classify the unknown data.

There are three main types of machine learning approaches: unsupervised, semi-supervised, and supervised. In unsupervised learning problems, the main task is to find patterns, structures, or knowledge in unlabeled data. When a portion of the data is labeled during acquisition of the data or by human experts, the problem is called semi-supervised learning. The addition of labeled data greatly helps to solve the problem. If the data are completely labeled, the problem is called supervised learning and generally the task is to find a function or model that explains the data. The approaches such as curve fitting or machine-learning methods are used to model the data to the underlying problem; the label is the problem variable. Recently, machine learning techniques have been used extensively in areas such as speech recognition [21], spam detection [18], computer vision [35], fraud detection [3], computer networks [5], and cyber intrusion detection [9], among others. The reason behind the popularity of machine learning is that it provides with a general framework to model and solve very complex problems. Furthermore, machine learning operates “on the fly” without requiring a model of the environment, the attacker’s behavior, and with (almost) no human intervention. These characteristics make machine learning the ideal choice to detect stealthy, dynamic and unpredictable cross-layer wireless security threats.

8.3 Taxonomy of Existing Wireless Network Attacks

In this section, we provide an overview of the relevant wireless network attacks to date. Although wireless attacks are diverse in nature, it is possible to classify them into two well-distinct classes. i.e., *active* and *passive* attacks. The two classes have the following features:

- *Active attacks*: these attacks aim at partially (or completely) altering, corrupting or destructing ongoing communications. Typical examples of such an attack are *Denial of Service* (DoS) attacks such as *jamming* where the *jammer* exploits the broadcast nature of wireless communications to intentionally generate interfering signals that can potentially block all ongoing communications over one or more wireless channels. Other classical and highly disrupting attacks are *selective forwarding* and *reply attacks*. The former attack is generally aimed at mesh and relay-aided networks where a malicious node can selectively drop or forward a subset of the received packets, thus generating a decrease in the packet delivery ratio of the network. Instead, replay attacks aim at retransmitting multiple copies of the same packet. While wired networks can efficiently avoid such an attack by means of firewalls, wireless networks are particularly vulnerable to replay attacks, especially when cloned packets corresponds to user association, handshakes and ACK packets.[1]

- *Passive attacks*: in contrast with active attackers, passive attackers do not interact with the system actively but, instead, limit their actions to the monitoring of ongoing communications. A classical example of such an attack is *eavesdropping*, where a malicious node, the so-called *eavesdropper* aims at monitoring wireless channels to detect ongoing communications. When a communication is detected, the eavesdropper can either get access to the content of transmitted packets and compromise the confidentiality of data, or it can monitor RF transmissions to obtain statistical information on routes, identity of nodes and their transmission activity.

It is clear that passive attackers are very hard to be detected as they never disclose their presence or position. Accordingly, to protect the network from passive attacks, prevention mechanisms such as encryption [56, 100], steganography [24, 25, 55, 57, 79] and access control lists [70] are generally employed. The network is generally not able to measure the effectiveness of any prevention mechanism. Therefore, prevention mechanisms are generally proactive and static.

For the above reasons, in this book chapter we focus our attention on active attacks only. That is, we will delve into those cases where it is possible to detect the presence of attackers, and it is possible to monitor and estimate the impact of their attacks on the achievable performance of the network.

8.4 Learning-Based Wireless Security Framework (LeWiS): An Overview

Figure 8.1 shows a block diagram of LeWiS. The core of LeWiS lies in two different yet interconnected modules, i.e., the learning-based network stack (LNS) and the learning-based control module (LCM).

Learning-Based Network Stack (LNS) The main task of the LNS is to adapt transmissions protocols and radio frequency (RF) configurations to implement dynamic protocols based on control logic algorithms defined in the LCM. To this end, the LNS (1) swiftly implements the necessary configuration changes to address adversarial action; (2) increases network throughput as much as possible; and (3) eases the definition of protocols with cross-layer optimum behaviors. The LNS operates on the whole set of existing networking layers, and involves protocols from PHY to MAC to routing layers and to transport layer.

[1] For a more detailed discussion on active attacks and their impact on wireless communications we refer the interested readers to [4, 85, 107], where an exhaustive analysis of active attacks and their corresponding defence mechanisms is provided.

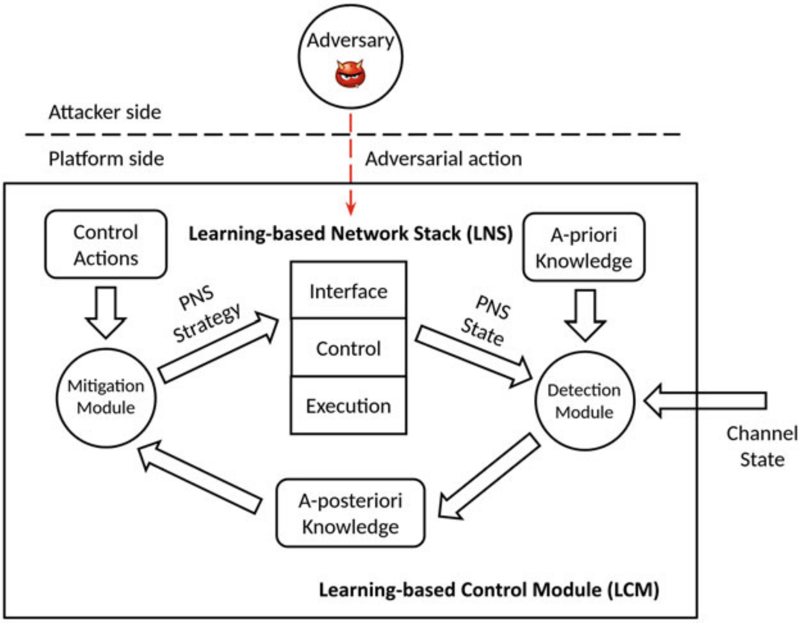


Fig. 8.1 Block scheme of the proposed Learning-based Wireless Security (LeWiS) framework

Although the logic of how network parameters are changed will be completely handled by the LCM, the LNS will be in charge of (1) actuating the commands received from the LCM; (2) coordinating the different network layers in a reliable and efficient way; and (3) storing and managing the current network state. Accordingly, the LNS is divided into three distinct but interacting "planes", each in charge of coordinating a different group of functionalities. We refer to these as the *execution* (where the protocols actually run), *control* (store network status information and receive parameters), and *interface* (provide a set of cross-platform APIs between the LCM and the LNS).

We now provide an overview of the LNS components and the interactions between them. We describe each component in a bottom-up fashion.

Execution Plane The execution plane (EP) handles the actual implementation of the networking algorithms. Specifically, it handles the set of data structures and algorithms, as well as the signal processing and RF front-end needed for wireless communication. As such, the EP is cross-layer and cross-domain by design, and resides on both the hardware (HW) and operating system (OS) portions of the wireless nodes.

The EP is structured as follows. The raw information bytes coming from the application (e.g., voice, video, data) are received, and after header formation at the transport (TSP), network (NET) and link (LNK) layers, the information packet is transformed into digital waveforms at the physical (PHY) layer and subsequently transmitted via the RF front-end. According to the commands provided by the control plane (CP), the EP chooses from a set of protocols and algorithms at every layer of the protocol stack. For example, the EP might choose different PHY schemes (e.g., OFDM, narrowband, spread-spectrum) or MAC schemes (e.g., CSMA, TDMA, ALOHA, etc.), according to what instructed by the CP.

The state of protocols at different layers are stored as variables in the CP. Therefore, any state variable of any protocol at any layer can be reconfigured on-the-fly through the CP. Protocol parameters (e.g., inter-frame spacing, modulation, carrier frequency, size of minimum contention window) can be "intrinsically" accessed by the CP as a result of a decision algorithm, or by the EP according to specific data acquisitions.

Control Plane The control plane (CP) is responsible for storing the network state composed by the set of parameters used at the different layers (e.g., modulation, transmit power, size of minimum CW, neighbouring list, and routing table). Furthermore, the CP is designed to handle the logic of data processing, which takes place in the EP. The CP decides the sequence of data processing functionalities that will be executed in the EP. The CP also controls switching between different protocols on-the-fly based on the decisions taken by the LeWiS mitigation module.

Interface Plane The interface plane provides a set of application program interface (API) between the LNS and the LCM. Specifically, the purpose of this layer is to (1) communicate to the LCM the state of the LNS held by the register and control plane; and (2) handle the reception of updated network parameters coming from the LCM. This plane can be implemented purely in software or also in hardware (e.g., FPGA), depending on the platform's needs.

Learning-Based Control Module (LCM) The purpose of the LCM is to implement the algorithms that will ultimately provide the LNS with appropriate network parameters at every layer of the protocol stack. The choice of network parameters will depend on (1) the current environmental conditions; and (2) the current network state, provided by the LNS. This information is used by the LCM to take appropriate action in response to various phenomena, ranging from a change in network objectives (e.g., more throughput is needed at the cost of energy consumption) to a detection of an ongoing network attack.

The LCM provides *decisions* based on user-defined machine learning algorithms. Such "decisions" may be different in nature; they include, for example, modifying a parameter in a protocol (e.g., transmit power), switching to a different transmission scheme at the PHY layer (e.g., from OFDM to CDMA), and complex cross-layer decision making such as joint routing and spectrum allocation, among others. The output of the LCM is not applied to the LNS directly, but specific APIs are used in the Interface plane to establish communication between the LCM with the LNS. Therefore, results of specific machine learning algorithms can be accessed by the Execution and Control planes of the LNS and adopted in their logic. Thus, *the LCM separates the decision plane from execution of the protocol stack, which enables the* *definition and reconfiguration of the decision logic on-the-fly without influencing the on-going protocol execution logic*. Most importantly, this separation provides the capability to (1) apply locally control decisions taken at other devices; and (2) create decisions to be applied at other network devices.

From a logical standpoint, the LCM is composed by a *detection* and a *mitigation* module. The detection module records a series of observed events and use them to update the a priori distribution of the parameters of the underlying network state. The resultant a posteriori distribution is fed to the mitigation module, which in turn decides the strategy to address the ongoing attack. Another task of the LCM is to sense the environment and acquire as much information as possible regarding the outside world. This includes carrier sensing, detection of number of nodes, and so on. This information will be fed to the machine learning algorithms, and if necessary used for training the models.

The rest of this section will be devoted to describe in details the LeWiS detection and mitigation modules. By doing so, we will also discuss the relevant state of the art on the topic of detection and mitigation of wireless attacks.

8.5 LeWiS Detection Module

In this section, we discuss the detection module of LeWiS. First, we introduce the general structure of the LeWiS detection module. Then, the challenges of detection are discussed. Next, we review different machine learning techniques that can be used in the LeWiS detection module. Finally, we discuss different detection approaches.

The objective of the detection module is to use machine learning techniques to construct reliable hypotheses on the existence of possible attacks in a network, based on observed evidences and *a-priori* knowledge on the attacks. The detection results are represented in the form of *a-posteriori* knowledge on the attacks and supplied to the mitigation module for defense strategy making.

The detection module is made up by three components, an event monitor, a learning engine, and a learning controller, as is shown in Fig. 8.2. We discuss each component below.

- *Event Monitor*. The event monitor observes and records events occurred on multiple layers that are useful for the detection of the underlying network state. Although an attack can be identified using the network state changes made by it, states are not always directly observable. On the other hand, there are always observable events whose occurrences are a function of the underlying network state. These event may include a successful (or failed) reception of a bit (or packet), the result of a channel contention, and the end-to-end delay of a message, among others. These events may reveal the underlying network state often in a *probabilistic* way.
- *Learning Engine*. The learning engine computes the probability that (1) a certain attack is taking place (supervised learning techniques), or (2) the belief that the network is running in a "normal" state (unsupervised learning techniques).

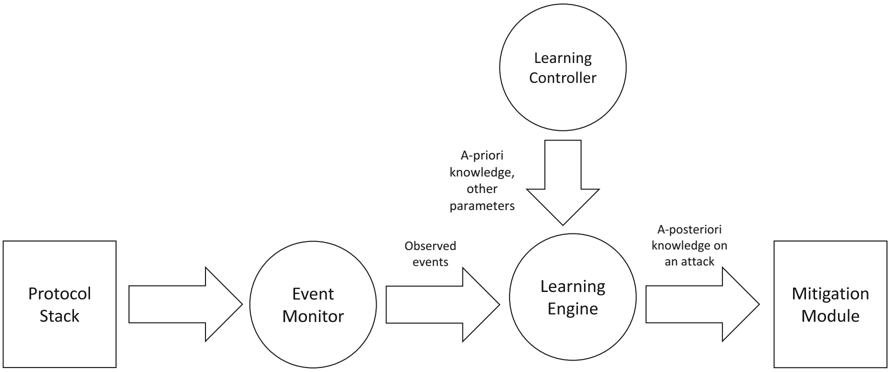


Fig. 8.2 Components of the detection module

Specifically, it takes the events observed by the event monitor as the input, generates a "data" point based on the events, which represents the network state. Then, learning techniques such as K nearest neighbor (KNN), support vector machine (SVM), or clustering are used upon the data point. In most cases, it also takes *a-priori* knowledge in some form as an input. The *a-priori* knowledge represents the knowledge on the expected behavior of an attack, and can be derived by theoretical analysis, or, more importantly, training.

- *Learning Controller*. The learning controller is mainly in charge of the parameter settings for the learning engine. Depending on the application and implementation of a network, it may be prone to different attacks. Meanwhile, depending on the threat of an attack, it may not always be worthy defending it, considering the overhead. This is especially important when the number of possible attacks are high. Therefore, the learning controller should decide which attacks to detect, and consequently, which layers and events to monitor. The learning controller is also in charge of feeding the learning engine with the proper *a-priori* knowledge.

8.5.1 Challenges in Cross-Layer Attack Detection

There are different ways to launch cross-layer attacks. "Sub-attacks" on different layers with similar consequences may be used *in parallel* for a better result. For example, observing that denial-of-service can be achieved using multiple attacks on different layers, it is argued in [81] that a cross-layer DoS attacks can be launched by jointly utilizing these sub-attacks. On the other hand, it is also possible for a sophisticated attacker to launch a more advanced cross-layer attack by exploiting the cross-layer interactions in the underlying protocols of the network (in the following contents, we will deliberately refer to this type of attacks as "advanced" cross-layer attacks when distinguishing is needed). Compared to traditional single-layer attacks, these advanced cross-layer attacks have several unique challenges:

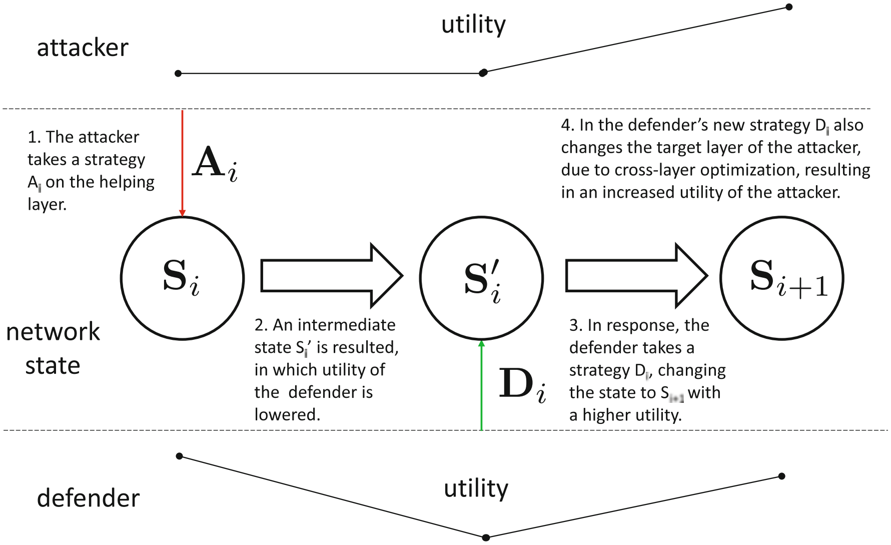

attacker
utility
1. The attacker takes a strategy A_i on the helping layer.
$\mathbf{A}_i$
4. In the defender's new strategy D_i also changes the target layer of the attacker, due to cross-layer optimization, resulting in an increased utility of the attacker.
$\mathbf{S}_i$
$\mathbf{S}'_i$
$\mathbf{S}_{i+1}$
network state
2. An intermediate state S_i' is resulted, in which utility of the defender is lowered.
$\mathbf{D}_i$
3. In response, the defender takes a strategy D_i, changing the state to S_i+1 with a higher utility.
defender
utility

Fig. 8.3 The mechanism of cross-layer attacks

- *Stealthiness*. The major issue in cross-layer attack detection is overcoming their stealthiness. To explain this point, let us consider the example shown in Fig. 8.3. We define as *network state* a state of network variables, which may include signal-to-interference-plus-noise ratio (SINR) of a link, the channel access probability of a node, and the quality of a route, among others. Let us suppose the network is in a state $\mathbf{S}_i$, and that the adversary aims at changing a certain *target* layer so that the network enters a new state where the defender is penalized. To this end, instead of attacking the target layer directly as in a traditional single-layer attack, in this case the adversary chooses another layer, i.e., the *helping* layer, and attacks it with a strategy $\mathbf{A}_i$. This causes the defender to switch to an intermediate state $\mathbf{S}'_i$ where the utility of the defender is lowered. As a response, the defender chooses a strategy $\mathbf{D}_i$ to switch to a more favorable state $\mathbf{S}_{i+1}$.

  Since the defender jointly optimizes its strategy on multiple layers, strategy $\mathbf{D}_i$ changes the target layer of the adversary as well, as long as the attack strategy $\mathbf{A}_i$ is carefully chosen. In other words, with a fine-tuned attack strategy, a cross-layer attack can create a state in which the defender's responsive strategy also benefits the adversary. Therefore, cross-layer attacks have the potential to be exceptionally *stealthy*, and how to detect the often small-scale attack activities becomes the most important challenge.
- *Heterogeneity*. Cross-layer attacks involve multiple "sub-attacks" on different layers. Therefore, different cross-layer attacks may behave similarly on a specific layer, especially on the helping layer. This suggests high heterogeneity in the form of attacks even if similar attack patterns have been observed.

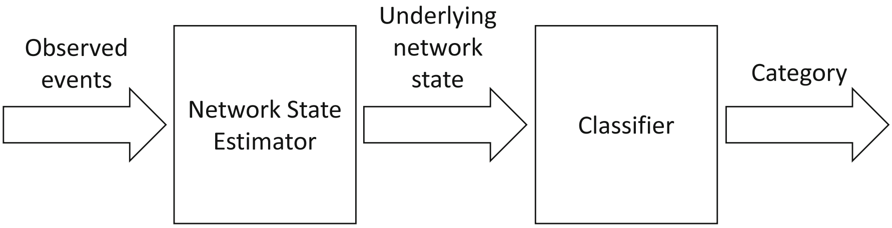


Fig. 8.4 Learning engine

The challenge of stealthiness suggests that the network state that distinguishes an attack is difficult to estimate; and the challenge of heterogeneity suggests that it is difficult to distinguish different attacks. In the learning engine, these challenges are addressed by two components, a network state estimator and a classifier, as shown in Fig. 8.4. The network state estimator estimates the underlying network state that may be ambiguous to a "normal" state due to small-scale attack activities, and generates a feature vector representing the current state. The classifier is trained to classify such a feature vector to a specific attack (misuse detection) or abnormal (anomaly detection).

8.5.2 Learning Techniques for Attack Detection

Various machine learning techniques can be applied to accomplish the tasks of both network state estimation and classification.

8.5.2.1 Network State Estimation Techniques

Network state estimator is tasked to estimate the underlying network state, which is usually not directly observable and may only be changed by the attacker slightly. It falls to the topic of parameter estimation in machine learning, and maximum likelihood (ML) estimation and Bayesian estimation are two widely-used methods [27].

ML Estimation ML estimation aims at finding the candidate hypothesis that maximizing the likelihood for the observed events to happen. Specifically, suppose the probability (likelihood) for an event $\mathbf{E}_k = \mathbf{e}_k$ to happen for the underlying network state $\mathbf{S} = \mathbf{s}$ is

$$\mathbb{P}\{\mathbf{E}_k = \mathbf{e}_k | \mathbf{S} = \mathbf{s}\} = f(\mathbf{e}_k, \mathbf{s}), \tag{8.1}$$

where the function $f(\cdot)$ is available by either theoretical analysis or training, then the ML estimator generates the hypothesis $\mathbf{S} = \mathbf{s}$ if

$$\mathbf{s} = \arg\max_{\mathbf{s}\in\mathscr{S}} f(\mathbf{e}_k, \mathbf{s}), \tag{8.2}$$

where $\mathscr{S}$ is the set of all possible values for the network state $\mathbf{S}$.

ML estimation treats the network state as a fixed value and finds the value best fits the evidences. It does not take into account the a-priori knowledge on the network state, which is usually available. Therefore, in such cases, it is not as good as Bayesian estimation, which is based on a-posteriori instead of likelihood. However, it is relatively less complex, especially when dealing with binary hypotheses. For example, log likelihood ratio test (LLR) is used for jamming detection [77].

Bayesian Estimation Bayesian estimation, on the other hand, treats the network state $\mathbf{S}$ as a random variable and constructs *a-posteriori* distribution. It takes into account the a-priori knowledge on the network state. Besides, as a distribution on the set of all possible values instead of the likelihood for one single value, Bayesian estimation provides more information on the network state than ML estimation. Therefore, for cross-layer attacks creating a network state $\mathbf{S}'$ that is only slightly different from the "normal" state $\mathbf{S}$, Bayesian estimation is generally a better choice than ML estimation.

Specifically, with a sequence of independent events $\{\mathbf{e}_k\}_{k=1,\ldots,K}$ have happened, it follows that

$$\mathbb{P}\{\{\mathbf{E}_k\}_{k=1,\ldots,K} = \{\mathbf{e}_k\}_{k=1,\ldots,K} | \mathbf{S} = \mathbf{s}\} = \prod_{k=1}^{K} f(\mathbf{e}_k, \mathbf{s}), \tag{8.3}$$

which leads to the following *a-posteriori* distribution

$$\begin{aligned} &\mathbb{P}\{\mathbf{S} = \mathbf{s} | \{\mathbf{E}_k\} = \{\mathbf{e}_k\}\} \\ &= \frac{\mathbb{P}\{\{\mathbf{E}_k\} = \{\mathbf{e}_k\} | \mathbf{S} = \mathbf{s}\}}{\int_{\mathbf{s}} \mathbb{P}\{\{\mathbf{E}_k\} = \{\mathbf{e}_k\} | \mathbf{S} = \mathbf{s}\} \cdot \mathbb{P}\{\mathbf{S} = \mathbf{s}\}\, \mathrm{d}\mathbf{s}} \cdot \mathbb{P}\{\mathbf{S} = \mathbf{s}\}, \end{aligned} \tag{8.4}$$

where $\mathbb{P}\{\mathbf{S} = \mathbf{s}\}$ is some *a-priori* distribution of $\mathbf{S}$, which represents the "normal" network state (i.e., without attacks). Note that such quantity is often available—for example, the distribution of SINR on a link can be derived from the fading model, the channel access probability for any node in a network running CSMA/CA is approximately the same, and so on. If accurate knowledge is not available, it is often possible to know some information on it, such as the functional form, and the range of its values [27].

The *a-posteriori* distribution in Eq. (8.4) reveals the possible underlying network state, so the defender is now aware of the attack *activities* of the attacker. The resultant distribution is fed to a classifier to classify a new data point. For example, with the signature of a suspected attack, we may have the classifier in the following general form:

$$C(\mathbb{P}\{\mathbf{S} = \mathbf{s} | \{\mathbf{E}_k\} = \{\mathbf{e}_k\}\}, \mathbf{F}) \underset{\text{Not attacked}}{\overset{\text{Attacked}}{\gtrless}} C_{\text{th}}, \tag{8.5}$$

where **F** represents the signature of the suspected attack, and C_{th} is a threshold that is set based on an *a-priori* training of the system. The detailed form of the classifier depends on the targeting attack, and the classification techniques. We will discuss the techniques in the following subsection.

8.5.2.2 Classification Techniques

There is a rich body of classification (or clustering for unsupervised learning) techniques used in network intrusion detection systems (NIDS). [45] gives an experimental comparison on such techniques applied to the KDD dataset [41]. KDD dataset is a famous benchmark dataset for network intrusion detection. It is packet-based, involving packets generated on network layer and above. The attacks are mainly single-layer attacks. However, the machine learning techniques still have the potential to be applied for the classification of cross-layer attacks. We have listed in Table. 8.1 the most widely-used techniques and the attacks they have been applied to.

K Nearest Neighbor (KNN) K Nearest Neighbor (KNN) is an instance-based learning technique, which classifies a data point based on the majority of the classes of its K nearest neighbors (usually according to the Euclidean distance in the feature space).

Table 8.1 Learning techniques used in attack detection

	Techniques	Attacks
Supervised	K nearest neighbors (KNN)	KDD attacks[a] [48], flooding [50]
	Decision tree	KDD attacks [73, 76]
	Support vector machine (SVM)	Sinking [37]
	Artificial neural network	KDD attacks [6, 53, 88], DDoS [67]
	Ensemble learning	KDD attacks [1, 32]
Unsupervised	Clustering	DDoS [47]
	K means	KDD attacks [52]
Reinforcement	Q-learning	Not specified [34, 72]

[a] We refer to KDD attacks the set of attacks in KDD dataset, including DoS, user to root (U2R), remote to local (R2L), and PROBE attacks

Formally, denoting the set of the K nearest data points as $\mathscr{K}$, then the probability that a new data point $\mathbf{x}$ belongs to a class y is computed as

$$\mathbb{P}\{y|\mathbf{x}\} = \frac{1}{K} \sum_{x_i \in \mathscr{K}} I(y_i = y), \tag{8.6}$$

where y_i is the label of data point x_i, and $I(y_i = y)$ is an indicator function for the condition $y_i = y$. The classifier is then assigns class y_* according to

$$y^* = \arg\max_{y \in \mathscr{Y}} \mathbf{P}\{y|\mathbf{x}\}, \tag{8.7}$$

with $\mathscr{Y}$ denoting the set of all classes.

KNN is one of the least complex machine learning techniques and thus has the potential to be applied in wireless networks with low cost devices. It has been evaluated on wireless sensor networks in [50] for flooding attack and used on the KDD dataset in [48].

Decision Tree Decision tree is a technique to organize multiple rules in a tree-like model. Each node in the tree except the leaves corresponds to a test over some attributes (or features) of a data point. The decision process is directed to different children based on the test result, until a leaf node is reached, which represents a decision. Decision tree techniques such as C4.5, random tree, and random forest have been applied to intrusion detection on the KDD dataset in [73, 76].

Support Vector Machine (SVM) Support Vector Machine is based on the assumption that data points for different classes are *separable* when represented in a high dimension feature space. The data points are first mapped to a high-dimensional feature space. Then, a linear decision function is constructed in the mapped feature space, resulting in hyperplanes separating two classes of data points.

Specifically, suppose the mapped data points is represented as a vector $\mathbf{x}$, then the classifier is in the form of

$$\mathbf{w}^T\mathbf{x} - b \geq 1 \tag{8.8}$$

for one class, and

$$\mathbf{w}^T\mathbf{x} - b \leq -1 \tag{8.9}$$

for the other. The weight $\mathbf{w}$ should be computed to maximize $\frac{2}{\|\mathbf{w}\|}$, i.e., the margin between the two classes.

In [37], SVM is used for the detection of sinking attack in wireless ad-hoc networks.

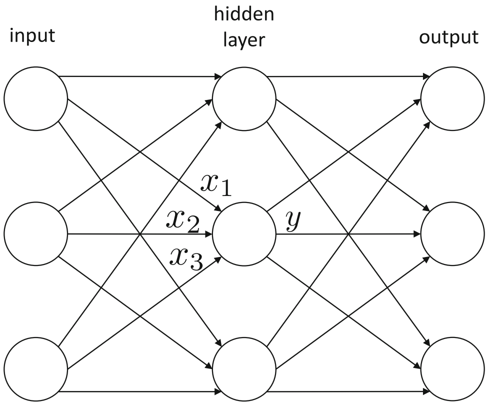


Fig. 8.5 Artificial neural network

Artificial Neural Network Artificial neural networks are inspired by biological neural networks. It is a network of artificial neurons (nodes), as shown in Fig. 8.5. "signals" are transmitted through connections between the neurons. At each neuron, the "signal" is processed using a (usually non-linear) function of the weighted sum of all inputs. Denoting the input as $\mathbf{x}$ and the weight as $\mathbf{w}$, the output y is

$$y = f(\mathbf{w}^T\mathbf{x}), \tag{8.10}$$

where the function $f(\cdot)$ may take different forms. There are typically multiple layers of neurons, with the first layer as the input and the last layer the output (the classification result). The weights in a neural network are trained to produce the most favorable outputs. It has found applications in detection of various attacks in [6, 53, 67, 90].

Ensemble Learning Ensemble learning techniques utilize the hypotheses generated by multiple weak learners to construct a strong one that outperforms each individual weak learner. There are multiple ways to ensemble weak learners, such as bagging and boosting [27]. In detection of cross-layer attacks, due to the heterogeneity in sub-attacks, the best learning techniques in detecting each of them may vary. Therefore, a good ensemble method may improve the performance of the detection. An AdaBoost-based learning with decision tree as weak learners is evaluated on KDD dataset in [32]. In [1] a simple ensemble method of weighted majority is used.

Clustering Clustering is an unsupervised learning technique that allows the unlabeled data points form different groups (clusters) automatically based on their features. Different methods can be used for this purpose. In [47] a hierarchical clustering method is used to detect DDoS attacks.

K Means K means is a classic clustering method. It first randomly creates K clusters and then iteratively updates the center of each cluster with new data points until convergence. New data points can be assigned to a cluster according to the distance.

For standard K means, suppose the mean (center) of cluster k at step i is $\mathbf{m}_k^i$, then a data point $\mathbf{x}_j$ is assigned to cluster

$$k^* = \arg\max_{1\leq k\leq K} \|\mathbf{x}_j - \mathbf{m}_k^i\|. \tag{8.11}$$

With all data points clustered, the new means are computed as

$$\mathbf{m}_k^{i+1} = \frac{1}{\|C_k^i\|} \sum_{j\in C_k^i} \mathbf{x}_j, \tag{8.12}$$

with C_k^i denoting the set of data points belonging to cluster k at step i. The update stops when there is only negligible changes in the means.

In [52] K means is used with KNN for intrusion detection, where K means is used to form clusters and KNN is used to assign new data points to the clusters.

Q-Learning Instead of aiming at classifying (or clustering) data points, reinforcement learning aims at finding the optimal policy to maximize the reward in a dynamic system with multiple states. Q-learning is a popular reinforcement learning technique, which can be implemented using dynamic programming. As a decision making process, it fits more for the task of attack mitigation (introduced with more details in Sect. 8.6), but there are still several works in applying Q-learning to attack detection [34, 72]. The common idea is that the detection and attack form a game, in which the defender and attacker may adopt different actions (to detect or not to detect for the defender; to attack or not to attack for the attacker) and get different rewards. Q-learning can be used to find the optimal policy for each side in this game-theoretical framework. Note this is a high-level model, and the actions for both sides are abstract, without detailed detection (or attack) methods involved, therefore, Q-learning (and reinforcement learning techniques in general) needs to be used with other detection methods dedicated for the attacks for a good performance.

8.5.2.3 Detection Approaches: Misuse vs Anomaly

We have reviewed different techniques that can be used in attack detection. An equally important question in detection module design is the detection approach. There are two approaches in intrusion detection, misuse detection and anomaly detection [42]. The former aims at identifying a specific attack based on its signature, while the latter aims ate finding out outliers to “normality”.

The framework established in [106] has adopted the misuse detection approach. It is argued that the learning engine should be tailored for each target attack individually. However, there are also scenarios where it is difficult to adopt this approach. For example, in networks with low cost devices and low security requirement, it is neither feasible nor necessary to monitor multiple advanced cross-layer attacks; in scenarios lacking knowledge on the potential attacks, it is also difficult to train a model for the unknown attack. In such cases, the anomaly detection approach seems more promising.

The common idea of anomaly detection is to monitor multi-layers and cluster the data points into different groups. Only one of the groups is considered normal, and for data points in other groups, the mitigation module will be called to "revert" the state back to normal. This step may contribute to the launching of an advanced cross-layer attack, as shown in step 3 in Fig. 8.3. Therefore, there is a possibility that anomaly detection fail against advanced cross-layer attacks.

Reference [87] is a typical example of this idea. It is assumed that there are intrinsic and observable distinction between normal and abnormal behaviors. The authors propose to select "features" that best distinguish normal and abnormal behaviors on multiple layers, and use classifiers such as decision tree, Bayesian Network, and SVM to classify if a data point is normal or not.

It is generally believed that anomaly detection is able to detect not only already-known attacks but also unknown attacks. However, in [75] it is argued that this may not hold in practice, due to the unsuitability to use machine learning in outlier detection, the high cost of errors, the semantic gap between the anomaly detection results and the insight on defense, among others. Therefore, anomaly detection may only enjoy the advantage of simplicity and low cost compared to misuse detection.

8.6 LeWiS Mitigation Module

The primary goal of the mitigation module is to efficiently counteract ongoing attacks by selecting and combining one or more defence strategies among a set of feasible defence mechanisms. In order to identify effective defence strategies, the mitigation module needs to address the following core challenges: (1) attacks can be launched by a potentially large number of adversaries; (2) the adversaries might be heterogeneous and be able to attack the network from multiple locations of the network; and (3) their behavior (i.e., their attack strategy, position, and so on) and number might vary over time.

The above challenges are peculiarly hard to be tackled. Indeed, it has been shown that attackers can easily hide their location by exploiting the broadcast nature of RF communications [26, 49], and can also maximize the undetectability and impact of their attacks by using simple but effective attacks. As an example,*pilot jamming* [15, 71] and *pilot nulling* [15] can be used to attack pilot tones and partially or completely corrupt synchronization and equalization operations at the receiver side. Another example is *reactive jamming* [25, 93], where the jammer uses energy-detector systems to first detect ongoing communications, and then transmit interfering signals aimed at disrupting RF transmissions.

This discussion shows that it is highly difficult (if not impossible) to fully characterize a network of attackers and deterministically derive the best defense strategy. Accordingly, not only is LeWiS tasked to learn how to defend the network from a particular attack, but it has also to adapt to the heterogeneous and ever-changing environment and attacks. Therefore, our framework LeWiS relies on a learning-based mitigation module where *a-priori* knowledge, real-time observations and supervised control actions are jointly leveraged to continuously adapt to network attacks and to derive appropriate defence mechanisms for any given network state and topology.

To provide the network with proper defense mechanisms, LeWiS continuously relies on the three components described below, and keeps tracking present and past network states. Also, the detection module notifies the mitigation module with relevant information on the nature of ongoing attacks. Accordingly, the mitigation module is capable of automatically computing the most effective defense strategy to mitigate, and possibly avoid, further attacks.

- *A-priori knowledge.* It is a set of static pre-loaded defence strategies for a given subset of attacks. It is used by the mitigation module to counteract ongoing attacks in the bootstrapping stage of LeWiS, i.e., when no information regarding the presence of attackers is available. Thus, as soon as an attack is detected, the mitigation module leverages the a-priori knowledge to identify one or more suitable defence strategies;
- *A-posteriori knowledge*: This set of strategies is continuously and autonomously built, updated and enhanced by LeWiS by evaluating the effectiveness of the different defence strategies employed in the past. Also, it is used by LeWiS to update the state of the network and to keep track of ongoing attacks. Indeed, it must be assumed that the environment and the attacker might change over time. As an example, a jammer can move from a location to another, of adapt its attack strategy to target a specific layer of the protocol stack. Therefore, the system must always monitor the environment to update the state and gather knowledge;
- *Supervised Control Actions*: The efficiency of the mitigation module can be improved by introducing human-generated inputs that either add new defence strategies to the system, or can steer the learning algorithm toward one solution rather than another. Specifically, it is possible for the network operator to interact with the decision mechanism by (1) specifying control operatives and performance metrics (e.g., throughput maximization, delay and energy minimization, minimum QoS guarantees, maximum transmission power, etc . . .); and (2) introducing new defence strategies when a novel attack is discovered and thus cannot yet be detected by LeWiS. These inputs are then used by LeWiS to identify the best defence strategy that optimizes a given control operative while satisfying a given set of constraints or requirements.

8.6.1 Traditional Mitigation Strategies

Given the cross-layer nature of attacks, LeWiS combines different defence strategies involving multiple layers of the protocol stack. As an example, to overcome reactive jamming attacks (i.e., the jammer is triggered when the received signal's power is above a given triggering threshold), defence strategies might involve the use of defense strategies at the crossroads of different layer. For example, power control (i.e., a physical layer strategy) can be used to reduce the strength of the signal received by the jammer, and channel access control (i.e., a link layer strategy) can be used to avoid the subset of channels monitored by the jammer [26].

To effectively address the variety of attacks, the mitigation module relies on a modular approach where atomic and single-layer defence strategies are selected and then properly combined to address different attacks. For the sake of clarity, in Table 8.2 we provide a brief summary of possible defences at each layer of the protocol stack.[2]

Table 8.2 Defence mechanisms and their application in wireless networks

	Physical layer	Data link layer	Network layer	Transport layer	Application layer
Power control	[26, 30, 89]				
Modulation	[98]				
Coding	[62]				
Beamforming	[29, 102]				
Friendly jamming	[33]				
Artificial noise	[62]				
Medium access control (MAC)		[26, 40, 97]			
Relaying and routing		[44, 62]	[39, 69]		
Authentication	[83, 101]	[40]	[7, 39]		[19]
Load balancing	[14]		[2]	[14]	
Transport layer security (TLS)				[66]	
Secure socket layers (SSL)				[66]	
Firewall					[94]
Encryption	[20, 54, 60, 61]	[17, 36]	[17, 54]	[17]	[94]

[2] The provided summary is not intended to be exhaustive, and a more detailed taxonomy of defense strategies can be found in [13, 96]. Furthermore, the same defence strategy (e.g., authentication, coding) can be implemented in different ways, and with possibly distinct outcomes, at multiple layers of the protocol stack.

The majority of prior work in network security has put its efforts on the design and implementation of ad hoc defence solutions for a small subset of network attacks that are targeted at the lower layers of the protocol stack [30, 39, 62]. On one hand, this deterministic approach produces highly reliable and robust solutions to a subset of well-defined attacks. On the other hand, it is clear that this paradigm cannot be equally productive in the case of cross-layer attacks where (1) the presence, position and attack strategy of the attackers is unavailable; and (2) different attacks can be combined together to attack multiple layers of the protocol stack.

The technological advancement and the availability of relatively cheap but powerful reprogrammable devices, such as SDRs and FPGAs, has enabled and facilitated the development and spread of complex and cross-layer attacks [12, 31, 106]. For this reason, ad hoc defence mechanisms addressing only a small number of attacks are expected to achieve poor performance—in other words, a single "one-fits-all" defense strategy is unlikely to be found. Instead, research efforts should be funneled towards the definition of a system capable of adaptively counteracting ongoing attacks through the generation of cross-layer strategies obtained by properly combining single-layer defense mechanisms. This latter research challenge will be the focus of the next section.

8.6.2 Learning-Aided Mitigation Techniques

To provide a reliable defence framework, information with respect to the attackers and their attack strategies is indeed required. Although many works in the literature assume that such an information is available to the defender, such requirement cannot be always satisfied in a significant number of wireless network scenarios where only incomplete and possibly erroneous information is available. To overcome the lack of information, statistical information and learning techniques can be profitably used to learn the environment, test different defense strategies and subsequently identify the most effective ones.

Since the system aims at providing a secure and tamper-proof communication, it is possible to model the mitigation module of LeWiS as an *agent* whose objective is to (1) select the most appropriate set of the defense strategies, such that (2) a given the security level and the performance of the network are simultaneously maximized over a large time horizon. In this context, the mitigation problem can be modeled as a *Markov Decision Process* (MDP) [92]. Specifically, the MDP corresponding to the mitigation module is shown in Fig. 8.6 and relies the four following fundamental elements:

- *Action Set*: it is represented by the set $\mathscr{A}$ of all feasible defense strategies. Each defence strategy $a \in \mathscr{A}$ is stored as single-layer atomic action that can be successively combined with other actions, belonging to the same or to different layers, to counteract cross-layer attacks;

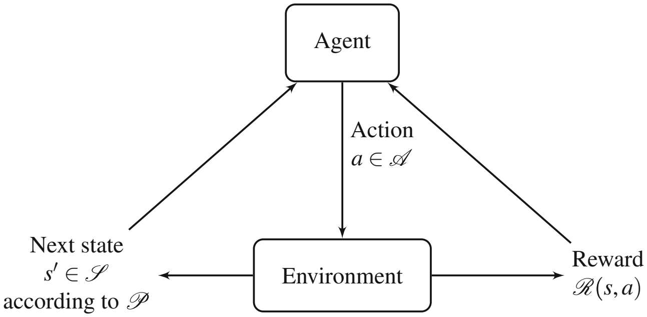


Fig. 8.6 Interactions in an MDP

- *State Set*: it represents the set $\mathscr{S}$ of possible states of the network, i.e., the network. As an example, each state $s \in \mathscr{S}$ can be used to represent network performance information such as bit, symbol and packet error rates, spectral efficiency and noise level. Furthermore, the state of the network also include information with respect to the type of attack, number of attackers and their position;
- *Reward Function*: this is a function $\mathscr{R} : \mathscr{S} \times \mathscr{A} \rightarrow \mathbb{R}$ that is used to represent the performance of the system when the network is in a given state and a given set of defence strategies are deployed. Typical examples of reward functions are throughput, data-rate, energy consumption and latency. The aforementioned metrics are used by the MM of LeWiS to evaluate the effectiveness of current defense strategies.
- *Transition Function*: a function used to regulate and describe the transition between a state to another. This function can be expressed as $\mathscr{P} : \mathscr{S} \times \mathscr{A} \times \mathscr{S} \rightarrow [0, 1]$ and represents the probability $\Pr\{s'|(a, s)\}$ that the network enters state s' when the action a is taken by the decision maker when the network is in state s.

ML technologies, have been profitably used in the literature to address a variety of cross-layer optimization problems ranging from rate maximization [16, 108], channel estimation [28, 86] and resource allocation [5, 80]. Preliminary works on the application of ML algorithms to address security issues in wireless networks are already available in the literature [5, 9, 65, 103], however they are generally targeted at mitigating only a limited subset of attacks, thus not providing effective and comprehensive solutions to counteract heterogeneous cross-layer attacks in wireless networks.

Given the potential of ML techniques that make it possible to learn from the environment and adapt defense strategies accordingly, we envision a cross-layer and comprehensive defense system where any attack can be mitigated, or possibly avoided, by wisely tuning the learning parameters of the system.

ML technologies can be generalized in two main classes, namely *Dynamic Programming* (DP) and *Reinforcement Learning* (RL), whose features are as follows:

- **Dynamic Programming**: this is a model-based approach where the impact of a given action on the transition from a state to another and the corresponding obtained reward is a known a-priori [11]. That is, DP requires knowledge of the reward and transition functions $\mathscr{R}$ and $\mathscr{P}$, respectively.
- **Reinforcement Learning**: this is a model-free approach where the learning process builds its own knowledge by means of observations and exploration of previous actions and rewards. That is, RL does not require any a-priori knowledge of $\mathscr{R}$ and $\mathscr{P}$ as the information related to the transition from a state to another and the achieved reward is discovered by the learning process itself [11].

The two above approaches have been widely and successfully used in the literature to derive optimal control policies for many networking problems. However, the above analysis clearly shows that DP require some form of knowledge with respect to the underlying MDP, a condition that might not be guaranteed in many network scenarios. On the contrary, RL iteratively constructs and gathers knowledge by exploring the environment and the action space, thus making it a promising technology to design a reliable and efficient MM when accurate information on the attacker is not available.

In the following, we present few examples that show how both DP and RL can be effectively used to counteract attacks in wireless networks.

8.6.2.1 DP and Its Application to Attack Mitigation

Dynamic programming is a well-established learning approach [8] that makes it possible to generally derive optimal solutions to a variety of NP-hard problems [95].

To compute efficient mitigation strategies, DP relies on the following well-known *Bellman's Equation*:

$$J_t(s_t) = \max_{a_t \in \mathscr{A}} \underbrace{\mathscr{R}_t(s_t, a_t)}_{\text{Single}-\text{slot Reward}} + \underbrace{\mathbb{E}_{\mathscr{P}}\{J_t(s_{t+1})|(s_t, a_t)\}}_{\text{Cumulative Expected Reward}} \tag{8.13}$$

As already outlined in the previous section, Eq. (8.13) shows that DP strongly depends on the state transition probabilities $\mathscr{P}$. Intuitively, the function $J_t(s_t)$ is iteratively maximized at each iteration by considering the best action a_t in the set $\mathscr{A}$ such that the single-slot reward $\mathscr{R}_t(s_t, a_t)$ and the cumulative expected future reward of network is maximized.

Apart from being dependent from the transition function $\mathscr{P}$, another drawback of DP lies in the so-called *curse of dimensionality* [8]. That is, to compute optimal defense strategies, DP generally needs to test a large number of actions (e.g., channel assignment, power allocation, encryption, beamforming, etc . . .), which eventually results in high computational complexity when the number of actions is large and heterogeneous. As an example, it has been shown [26] that even a relatively simple cross-layer defense strategy that merges physical and data-link layer defense mechanisms, e.g., power control and channel allocation, generally results in NP-hard solutions. Specifically, to counteract reactive jamming attacks, DP requires exponential time to compute an effective defense strategy even if the continuous transmission power levels are quantized and substituted with their corresponding discrete variables. To overcome such an issue, a promising approach is to leverage on the learning engine of DP and exclude inefficient defense strategies from the action set [26]. As an example, if a given defence strategy is known to be either nonfunctional or inefficient, e.g., the network is aware that all transmission power levels above a given triggering threshold always activate the jammer, those actions can be removed from the action set. Such an approach not only reduces the complexity of the overall learning algorithm, but it also avoids poor performance due the deployment of ineffective defense strategies.

8.6.2.2 RL and Its Application to Attack Mitigation

In the context of RL many algorithms have been proposed to derive optimal and sub-optimal policies for a variety of optimization problems [10, 38, 78]. Those algorithms have different properties and provide different performance levels in different network scenarios. For the sake of simplicity, in this chapter we will focus our attention on two well-known and well-established RL algorithms, namely *Q-Learning* and *State–action–reward–state–action*.

- **Q-Learning** : this learning approach relies on the well-known *Q-function*. Let a_t and s_t be the action taken by the agent and the state of the system at time t, respectively. The Q-function is defined as follows:

$$Q(s_t, a_t) = (1 - \alpha_t(s_t, a_t))Q(s_t, a_t) + \alpha_t(s_t, a_t)\left[\mathscr{R}_t + \gamma \max_{a^* \in \mathscr{A}} Q(s_{t+1}, a^*)\right] \tag{8.14}$$

  where $\alpha_t(s_r, a_t) \in [0, 1)$ is the *learning rate* of the algorithm associated to the 2-tuple (a_t, s_t), and $\gamma \in (0, 1)$ is the *discount parameter* and s_{t+1} is the observed state of the network when action a_t was taken. Intuitively, (8.14) iteratively aims at maximizing the total discounted reward of the network, which in our case consists in the maximization of network performance through the mitigation of network attacks. It is worth mentioning that there are no particular restrictions on the action selection mechanism to be used in the learning process. Well-established and widely used action selection mechanisms are random selection, ε-greedy and softmax algorithms [78]. However, other approaches are available, and they can be found in [82]. It has been shown [91] that if the reward function is bounded and $\sum_{t=1}^{+\infty} \alpha_t^2(s_t, a_t) < \sum_{t=1}^{+\infty} \alpha_t(s_r, a_t) = +\infty$ for all $(s_t, a_t) \in \mathscr{S} \times \mathscr{A}$, then (8.14) converges towards an optimal value of the Q-function when $t \rightarrow +\infty$. That is, if all the 2-tuples $(s_t, a_t) \in \mathscr{S} \times \mathscr{A}$ are visited infinitely often, the total discounted reward is guaranteed to converge to an optimal value.

- **State–action–reward–state–action (SARSA)** : this learning approach is similar to Q-learning but differs on how the Q-function is updated at each iteration. Specifically, SARSA relies on the following iterative equation:

$$Q(s_t, a_t) = (1 - \alpha_t(s_t, a_t))Q(s_t, a_t) + \alpha_t(s_t, a_t)\left[\mathscr{R}_t + \gamma Q(s_{t+1}, a_{t+1})\right] \tag{8.15}$$

where the only difference with the Q-learning approach is that the latter updates the value of (8.14) by computing the best action such that $a^* = \arg\max_{a \in \mathscr{A}} Q_t(s, a)$. Instead, in (8.15) the algorithm utilizes the same action selection mechanism at each iteration t. That is, while Q-learning uses the optimal action a^* to update the value of (8.14), SARSA uses the same selection algorithm, e.g., ε-greedy, to compute each action a_t and directly uses it to update the Q-function in (8.15). In general, SARSA has lower-complexity if compared to Q-learning, however it often produces only near-optimal improvements at each iteration of the learning process.

8.7 Conclusions

With such massive presence of interconnected wireless nodes deployed all around us, there are still exciting yet significant security research challenges that need to be addressed in the upcoming years. In this chapter, we have provided our perspective on the issue of cross-layer wireless security, which is based on a unique mixture of machine learning and software-defined radios. Specifically, we have introduced and discussed a Learning-based Wireless Security (LeWiS) framework that provides a closed-loop approach to the problem of cross-layer wireless security by leveraging machine learning and software-defined radios. We have first provided a brief review of background notions in Sect. 8.2, followed by an in depth-discussion of the LeWiS framework and its components. We have categorized and summarized the relevant state-of-the-art research. We hope that this work will inspire fellow researchers to investigate topics pertaining to cross-layer wireless security and keep in the race for a more secure technological world.

Acknowledgements This work is based upon work supported in part by ONR grants 0014-16-1-2213 and N00014-17-1-2046, ARMY W911NF-17-1-0034, and NSF CNS-1618727.

References

1. A.A. Aburomman, M.B.I. Reaz, A novel svm-knn-pso ensemble method for intrusion detection system. Appl. Soft Comput. **38**, 360–372 (2016)
2. A.A. Ahmed, N.F. Fisal, Secure real-time routing protocol with load distribution in wireless sensor networks. Secur. Commun. Netw. **4**(8), 839–869 (2011)

3. L. Akoglu, C. Faloutsos, Anomaly, event, and fraud detection in large network datasets, in *Proceeding of the ACM International Conference on Web Search and Data Mining (WSDM)* (2013), pp. 773–774
4. I.F. Akyildiz, X. Wang, A survey on wireless mesh networks. IEEE Commun. Mag. **43**(9), S23–S30 (2005)
5. M.A. Alsheikh, S. Lin, D. Niyato, H.P. Tan, Machine learning in wireless sensor networks: algorithms, strategies, and applications. IEEE Commun. Surv. Tutorials **16**(4), 1996–2018 (2014)
6. R.A.R. Ashfaq, X.Z. Wang, J.Z. Huang, H. Abbas, Y.L. He, Fuzziness based semi-supervised learning approach for intrusion detection system. Inf. Sci. **378**, 484–497 (2017). https://doi.org/10.1016/j.ins.2016.04.019. http://www.sciencedirect.com/science/article/pii/S0020025516302547
7. D. Balfanz, D.K. Smetters, P. Stewart, H.C. Wong, Talking to strangers: authentication in ad-hoc wireless networks, in *NDSS* (2002)
8. D.P. Bertsekas, *Dynamic Programming and Optimal Control*, vol. 1 (Athena Scientific, Belmont, MA, 1995)
9. A.L. Buczak, E. Guven, A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutorials **18**(2), 1153–1176 (2016)
10. L. Busoniu, R. Babuska, B. De Schutter, A comprehensive survey of multiagent reinforcement learning. IEEE Trans. Syst. Man Cybern. C **38**(2), 156–172 (2008)
11. L. Buşoniu, B. De Schutter, R. Babuška, Approximate dynamic programming and reinforcement learning, in *Interactive Collaborative Information Systems* (Springer, New York, 2010), pp. 3–44
12. A. Cassola, W.K. Robertson, E. Kirda, G. Noubir, A practical, targeted, and stealthy attack against wpa enterprise authentication, in *NDSS* (2013)
13. X. Chen, K. Makki, K. Yen, N. Pissinou, Sensor network security: a survey. IEEE Commun. Surv. Tutorials **11**(2), 52–73 (2009)
14. M. Chiang, Balancing transport and physical layers in wireless multihop networks: jointly optimal congestion control and power control. IEEE J. Sel. Areas Commun. **23**(1), 104–116 (2005)
15. T.C. Clancy, Efficient ofdm denial: pilot jamming and pilot nulling, in *IEEE International Conference on Communications (ICC)* (IEEE, Kyoto, 2011), pp. 1–5
16. C. Clancy, J. Hecker, E. Stuntebeck, T. O'Shea, Applications of machine learning to cognitive radio networks. IEEE Wirel. Commun. **14**(4), 47–52 (2007)
17. S. Convery, *Network Security Architectures* (Pearson Education, Chennai, 2004)
18. M. Crawford, T.M. Khoshgoftaar, J.D. Prusa, A.N. Richter, H.A. Najada, Survey of review spam detection using machine learning techniques. J. Big Data **2**(1), 2–23 (2015)
19. M.L. Das, Two-factor user authentication in wireless sensor networks. IEEE Trans. Wirel. Commun. **8**(3), 1086–1090 (2009)
20. T.R. Dean, A.J. Goldsmith, Physical-layer cryptography through massive mimo. IEEE Trans. Inf. Theory **63**(8), 5419–5436 (2017). https://doi.org/10.1109/TIT.2017.2715187
21. L. Deng, G. Hinton, B. Kingsbury, New types of deep neural network learning for speech recognition and related applications: an overview, in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)* (2013), pp. 8599–8603
22. L. Deng, D. Yu et al., Deep learning: methods and applications. Found. Trends Signal Process. **7**(3–4), 197–387 (2014)
23. L. Ding, T. Melodia, S. Batalama, J. Matyjas, M. Medley, Cross-layer routing and dynamic spectrum allocation in cognitive radio ad hoc networks. IEEE Trans. Veh. Technol. **59**, 1969–1979 (2010)
24. S. D'Oro, L. Galluccio, G. Morabito, S. Palazzo, Efficiency analysis of jamming-based countermeasures against malicious timing channel in tactical communications, in *2013 IEEE International Conference on Communications (ICC)* (2013), pp. 4020–4024. https://doi.org/10.1109/ICC.2013.6655188

25. S. D'Oro, L. Galluccio, G. Morabito, S. Palazzo, L. Chen, F. Martignon, Defeating jamming with the power of silence: a game-theoretic analysis. IEEE Trans. Wirel. Commun. **14**(5), 2337–2352 (2015)
26. S. D'Oro, E. Ekici, S. Palazzo, Optimal power allocation and scheduling under jamming attacks. IEEE/ACM Trans. Netw. **25**(3), 1310–1323 (2017). https://doi.org/10.1109/TNET.2016.2622002
27. R.O. Duda, P.E. Hart, D.G. Stork, *Pattern Classification*, 2nd edn. (Wiley, New York, 2000)
28. E.F. Flushing, J. Nagi, G.A. Di Caro, A mobility-assisted protocol for supervised learning of link quality estimates in wireless networks, in *International Conference on Computing, Networking and Communications (ICNC)* (IEEE, Maui, 2012), pp. 137–143
29. A.B. Gershman, U. Nickel, J.F. Bohme, Adaptive beamforming algorithms with robustness against jammer motion. IEEE Trans. Signal Process. **45**(7), 1878–1885 (1997)
30. P.K. Gopala, L. Lai, H. El Gamal, On the secrecy capacity of fading channels. IEEE Trans. Inf. Theory **54**(10), 4687–4698 (2008)
31. K. Hasan, S. Shetty, T. Oyedare, Cross layer attacks on gsm mobile networks using software defined radios, in *2017 14th IEEE Annual Consumer Communications Networking Conference (CCNC)* (2017), pp. 357–360. https://doi.org/10.1109/CCNC.2017.7983134
32. W. Hu, W. Hu, S. Maybank, Adaboost-based algorithm for network intrusion detection. IEEE Trans. Syst. Man Cybern. B Cybern. **38**(2), 577–583 (2008)
33. J. Huang, A.L. Swindlehurst, Cooperative jamming for secure communications in mimo relay networks. IEEE Trans. Signal Process. **59**(10), 4871–4884 (2011)
34. J.Y. Huang, I.E. Liao, Y.F. Chung, K.T. Chen, Shielding wireless sensor network using Markovian intrusion detection system with attack pattern mining. Inf. Sci. **231**, 32–44 (2013). https://doi.org/10.1016/j.ins.2011.03.014. http://www.sciencedirect.com/science/article/pii/S0020025511001435. Data Mining for Information Security
35. Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, T. Darrell, Caffe: convolutional architecture for fast feature embedding, in *Proceedings of the ACM International Conference on Multimedia* (2014), pp. 675–678
36. I. Jirón, I. Soto, R. Carrasco, N. Becerra, Hyperelliptic curves encryption combined with block codes for Gaussian channel. Int. J. Commun. Syst. **19**(7), 809–830 (2006)
37. J.F.C. Joseph, B.S. Lee, A. Das, B.C. Seet, Cross-layer detection of sinking behavior in wireless ad hoc networks using svm and fda. IEEE Trans. Dependable Secure Comput. **8**(2), 233–245 (2011)
38. L.P. Kaelbling, M.L. Littman, A.W. Moore, Reinforcement learning: a survey. J. Artif. Intell. Res. **4**, 237–285 (1996)
39. C. Karlof, D. Wagner, Secure routing in wireless sensor networks: attacks and countermeasures. Ad Hoc Netw. **1**(2–3), 293–315 (2003)
40. C. Karlof, N. Sastry, D. Wagner, Tinysec: a link layer security architecture for wireless sensor networks, in *Proceedings of the 2nd International Conference on Embedded Networked Sensor Systems* (ACM, New York, 2004), pp. 162–175
41. Kdd cup 1999 data (1999). http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html
42. R.A. Kemmerer, G. Vigna, Intrusion detection: a brief history and overview. Computer **35**(4), 27–30 (2002). https://doi.org/10.1109/MC.2002.1012428
43. D. Kreutz, F.M.W. Ramos, P.E. Verissimo, C.E. Rothenberg, S. Azodolmolky, S. Uhlig, Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015). https://doi.org/10.1109/JPROC.2014.2371999
44. I. Krikidis, J.S. Thompson, S. McLaughlin, Relay selection for secure cooperative networks with jamming. IEEE Trans. Wirel. Commun. **8**(10), 5003–5011 (2009)
45. P. Laskov, P. Düssel, C. Schäfer, K. Rieck, Learning intrusion detection: supervised or unsupervised?, in *Image Analysis and Processing – ICIAP 2005*, ed. by F. Roli, S. Vitulano (Springer, Berlin/Heidelberg, 2005), pp. 50–57
46. Y. LeCun, Y. Bengio, G. Hinton, Deep learning. Nature **521**(7553), 436 (2015)
47. K. Lee, J. Kim, K.H. Kwon, Y. Han, S. Kim, Ddos attack detection method using cluster analysis. Exp. Syst. Appl. **34**(3), 1659–1665 (2008)

48. Y. Li, L. Guo, An active learning based tcm-knn algorithm for supervised network intrusion detection. Comput. Secur. **26**(7), 459–467 (2007)
49. M. Li, I. Koutsopoulos, R. Poovendran, Optimal jamming attacks and network defense policies in wireless sensor networks, in *26th IEEE International Conference on Computer Communications. INFOCOM 2007* (IEEE, Alaska, 2007), pp. 1307–1315
50. W. Li, P. Yi, Y. Wu, L. Pan, J. Li, A new intrusion detection system based on knn classification algorithm in wireless sensor network. J. Elect. Comput. Eng. **2014**, Article ID 240217 (2014)
51. X. Lin, N.B. Shroff, R. Srikant, A tutorial on cross-layer optimization in wireless networks. IEEE J. Sel. Areas Commun. **24**, 1452–1463 (2006)
52. W.C. Lin, S.W. Ke, C.F. Tsai, Cann: an intrusion detection system based on combining cluster centers and nearest neighbors. Knowl. Based Syst. **78**, 13–21 (2015)
53. G. Liu, Z. Yi, S. Yang, A hierarchical intrusion detection model based on the pca neural networks. Neurocomputing **70**(7), 1561–1568 (2007). https://doi.org/10.1016/j.neucom.2006.10.146. http://www.sciencedirect.com/science/article/pii/S0925231206004644. Advances in Computational Intelligence and Learning
54. S. Liu, Y. Hong, E. Viterbo, Unshared secret key cryptography. IEEE Trans. Wirel. Commun. **13**(12), 6670–6683 (2014)
55. J. Lubacz, W. Mazurczyk, K. Szczypiorski, Principles and overview of network steganography. IEEE Commun. Mag. **52**(5), 225–229 (2014)
56. W. Mao, *Modern Cryptography: Theory and Practice* (Prentice Hall, Upper Saddle River, 2003)
57. D. Martins, H. Guyennet, Steganography in mac layers of 802.15. 4 protocol for securing wireless sensor networks, in *2010 International Conference on Multimedia Information Networking and Security (MINES)* (IEEE, Nanjing, 2010), pp. 824–828
58. T. Melodia, H. Kulhandjian, L.C. Kuo, E. Demirors, *Advances in Underwater Acoustic Networking* (Wiley, New York, 2013), pp. 804–852. https://doi.org/10.1002/9781118511305.ch23. http://dx.doi.org/10.1002/9781118511305.ch23
59. R.F. Molanes, J.J. Rodríguez-Andina, J. Fariña, Performance characterization and design guidelines for efficient processor - fpga communication in cyclone v fpsocs. IEEE Trans. Ind. Electron. **65**(5), 4368–4377 (2018). https://doi.org/10.1109/TIE.2017.2766581
60. L. Mucchi, L.S. Ronga, E. Del Re, A novel approach for physical layer cryptography in wireless networks. Wirel. Pers. Commun. **53**(3), 329–347 (2010)
61. L. Mucchi, L.S. Ronga, E. Del Re, Physical layer cryptography and cognitive networks, in *Trustworthy Internet* (Springer, New York, 2011), pp. 75–91
62. A. Mukherjee, S.A.A. Fakoorian, J. Huang, A.L. Swindlehurst, Principles of physical layer security in multiuser wireless networks: a survey. IEEE Commun. Surv. Tutorials **16**(3), 1550–1573 (2014)
63. M.D.V. Pena, J.J. Rodriguez-Andina, M. Manic, The internet of things: the role of reconfigurable platforms. IEEE Ind. Electron. Mag. **11**(3), 6–19 (2017). https://doi.org/10.1109/MIE.2017.2724579
64. S. Pudlewski, N. Cen, Z. Guan, T. Melodia, Video transmission over lossy wireless networks: a cross-layer perspective. IEEE J. Sel. Top. Signal Process. **9**, 6–22 (2015)
65. O. Puñal, I. Aktaş, C.J. Schnelke, G. Abidin, K. Wehrle, J. Gross, Machine learning-based jamming detection for ieee 802.11: design and experimental evaluation, in *2014 IEEE 15th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (IEEE, Sydney, 2014), pp. 1–10
66. E. Rescorla, *SSL and TLS: Designing and Building Secure Systems*, vol. 1 (Addison-Wesley, Reading, 2001)
67. A. Saied, R.E. Overill, T. Radzik, Detection of known and unknown ddos attacks using artificial neural networks. Neurocomputing **172**, 385–393 (2016)
68. A.L. Samuel, Some studies in machine learning using the game of checkers. IBM J. Res. Dev. **44**(1–2), 206–226 (2000)
69. K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, E.M. Belding-Royer, A secure routing protocol for ad hoc networks, in *10th IEEE International Conference on Network Protocols, 2002. Proceedings* (IEEE, Paris, 2002), pp. 78–87

70. N. Sastry, D. Wagner, Security considerations for ieee 802.15. 4 networks, in *Proceedings of the 3rd ACM workshop on Wireless Security* (ACM, New York, 2004), pp. 32–42
71. C. Shahriar, S. Sodagari, T.C. Clancy, Performance of pilot jamming on mimo channels with imperfect synchronization, in *2012 IEEE International Conference on Communications (ICC)*, pp. 898–902 (IEEE, Ottawa, 2012)
72. S. Shamshirband, A. Patel, N.B. Anuar, M.L.M. Kiah, A. Abraham, Cooperative game theoretic approach using fuzzy q-learning for detecting and preventing intrusions in wireless sensor networks. Eng. Appl. Artif. Intell. **32**, 228–241 (2014)
73. S.S.S. Sindhu, S. Geetha, A. Kannan, Decision tree based light weight intrusion detection using a wrapper approach. Exp. Syst. Appl. **39**(1), 129–141 (2012). https://doi.org/10.1016/j.eswa.2011.06.013. http://www.sciencedirect.com/science/article/pii/S0957417411009080
74. K.J. Singh, D.S. Kapoor, Create your own internet of things: a survey of IoT platforms. IEEE Consum. Electron. Mag. **6**(2), 57–68 (2017)
75. R. Sommer, V. Paxson, Outside the closed world: on using machine learning for network intrusion detection, in *2010 IEEE Symposium on Security and Privacy* (2010), pp. 305–316
76. G. Stein, B. Chen, A.S. Wu, K.A. Hua, Decision tree classifier for network intrusion detection with ga-based feature selection, in *Proceedings of the 43rd Annual Southeast Regional Conference - Volume 2, ACM-SE 43* (ACM, New York, 2005), pp. 136–141. https://doi.org/10.1145/1167253.1167288. http://doi.acm.org/10.1145/1167253.1167288
77. M. Strasser, B. Danev, S. Čapkun, Detection of reactive jamming in sensor networks. ACM Trans. Sens. Netw. **7**(2), 16:1–16:29 (2010)
78. R.S. Sutton, A.G. Barto, *Reinforcement Learning: An Introduction*, vol. 1 (MIT Press, Cambridge, 1998)
79. K. Szczypiorski, Hiccups: hidden communication system for corrupted networks, in *International Multi-Conference on Advanced Computer Systems* (2003), pp. 31–40
80. A. Testolin, M. Zanforlin, M.D.F. De Grazia, D. Munaretto, A. Zanella, M. Zorzi, M. Zorzi, A machine learning approach to qoe-based video admission control and resource allocation in wireless systems, in *2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)* (IEEE, Piran, 2014), pp. 31–38
81. G. Thamilarasu, A. Balasubramanian, S. Mishra, R. Sridhar, A cross-layer based intrusion detection approach for wireless ad hoc networks, in *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*, Washington (2005)
82. A.D. Tijsma, M.M. Drugan, M.A. Wiering, Comparing exploration strategies for q-learning in random stochastic mazes, in *2016 IEEE Symposium Series on Computational Intelligence (SSCI)* (2016), pp. 1–8. https://doi.org/10.1109/SSCI.2016.7849366
83. O. Ureten, N. Serinken, Wireless security through rf fingerprinting. Can. J. Elect. Comput. Eng. **32**(1), 27–33 (2007)
84. S. Vadlamani, B. Eksioglu, H. Medal, A. Nandi, Jamming attacks on wireless networks: a taxonomic survey. Int. J. Prod. Econ. **172**, 76–94 (2016)
85. J.P. Walters, Z. Liang, W. Shi, V. Chaudhary, Wireless sensor network security: a survey, in *Security in Distributed, Grid, Mobile, and Pervasive Computing*, vol. 1 (Auerbach, Boston, 2007), p. 367
86. Y. Wang, M. Martonosi, L.S. Peh, Predicting link quality using supervised learning in wireless sensor networks. ACM SIGMOBILE Mob. Comput. Commun. Rev. **11**(3), 71–83 (2007)
87. X. Wang, J.S. Wong, F. Stanley, S. Basu, Cross-layer based anomaly detection in wireless mesh networks, in *Proceedings of International Symposium on Applications and the Internet*, Bellevue (2009), pp. 9–15
88. S.S. Wang, K.Q. Yan, S.C. Wang, C.W. Liu, An integrated intrusion detection system for cluster-based wireless sensor networks. Exp. Syst. Appl. **38**(12), 15234–15243 (2011)
89. X. Wang, M. Tao, J. Mo, Y. Xu, Power and subcarrier allocation for physical-layer security in ofdma-based broadband wireless networks. IEEE Trans. Inf. Forensics Secur. **6**(3), 693–702 (2011)
90. X. Wang, Q.Z. Sheng, X.S. Fang, L. Yao, X. Xu, X. Li, An integrated Bayesian approach for effective multi-truth discovery, in *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management* (ACM, New York, 2015), pp. 493–502

91. C.J. Watkins, P. Dayan, Q-learning. Mach. Learn. **8**(3–4), 279–292 (1992)
92. D.J. White, *Markov Decision Processes* (Wiley, New York, 1993)
93. M. Wilhelm, I. Martinovic, J.B. Schmitt, V. Lenders, Short paper: reactive jamming in wireless networks: how realistic is the threat?, in *Proceedings of the Fourth ACM Conference on Wireless Network Security* (ACM, New York, 2011), pp. 47–52
94. M. Wilhelm, I. Martinovic, J.B. Schmitt, V, Lenders, Wifire: a firewall for wireless networks, in *ACM SIGCOMM Computer Communication Review*, vol. 41 (ACM, New York, 2011), pp. 456–457
95. G.J. Woeginger, Exact algorithms for np-hard problems: a survey, in *Combinatorial Optimization – Eureka, You Shrink!* (Springer, New York, 2003), pp. 185–207
96. B. Wu, J. Chen, J. Wu, M. Cardei, A survey of attacks and countermeasures in mobile ad hoc networks, in *Wireless Network Security* (Springer, New York, 2007), pp. 103–135
97. Y. Wu, B. Wang, K.R. Liu, T.C. Clancy, Anti-jamming games in multi-channel cognitive radio networks. IEEE J. Sel. Areas Commun. **30**(1), 4–15 (2012)
98. F. Wu, R. Zhang, L.L. Yang, W. Wang, Transmitter precoding-aided spatial modulation for secrecy communications. IEEE Trans. Veh. Technol. **65**(1), 467–471 (2016)
99. A.M. Wyglinski, D.P. Orofino, M.N. Ettus, T.W. Rondeau, Revolutionizing software defined radio: case studies in hardware, software, and education. IEEE Commun. Mag. **54**(1), 68–75 (2016)
100. Z. Xi, L. Li, G. Shi, S. Wang, A comparative study of encryption algorithms in wireless sensor network, in *Wireless Communications, Networking and Applications*, ed. by Q.A. Zeng (Springer, New Delhi, 2016), pp. 1087–1097
101. L. Xiao, L. Greenstein, N. Mandayam, W. Trappe, Fingerprints in the ether: using the physical layer for wireless authentication, in *IEEE International Conference on Communications, 2007. ICC'07* (IEEE, Glasgow, 2007), pp. 4646–4651
102. Q. Yan, H. Zeng, T. Jiang, M. Li, W. Lou, Y.T. Hou, Mimo-based jamming resilient communication in wireless networks, in *2014 Proceedings IEEE INFOCOM* (IEEE, Toronto, 2014), pp. 2697–2706
103. Z. Yu, J.J. Tsai, A framework of machine learning based intrusion detection for wireless sensor networks, in *IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC'08* (IEEE, Taichung, 2008), pp. 272–279
104. L. Zhang, T. Melodia, Hammer and anvil: the threat of a cross-layer jamming-aided data control attack in multihop wireless networks, in *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, Florence (2015), pp. 361–369
105. L. Zhang, F. Restuccia, T. Melodia, S.M. Pudlewski, Learning to detect and mitigate cross-layer attacks in wireless networks: framework and applications, in *Proceedings of the IEEE Conference on Communications and Network Security (CNS)*, Las Vegas (2017), pp. 361–369
106. L. Zhang, F. Restuccia, T. Melodia, S. Pudlewski, Taming cross-layer attacks in wireless networks: a Bayesian learning approach. IEEE Trans. Mobile Comput. (2018). https://doi.org/10.1109/TMC.2018.2864155
107. J. Zheng, A. Jamalipour, *Wireless Sensor Networks: A Networking Perspective* (Wiley, New York, 2009)
108. P. Zhou, Y. Chang, J.A. Copeland, Reinforcement learning for repeated power control game in cognitive radio networks. IEEE J. Sel. Areas Commun. **30**(1), 54–69 (2012)
109. L. Zhou, D. Wu, B. Zheng, M. Guizani, Joint physical-application layer security for wireless multimedia delivery. IEEE Commun. Mag. **52**(3), 66–72 (2014). https://doi.org/10.1109/MCOM.2014.6766087
110. Y. Zou, J. Zhu, X. Wang, L. Hanzo, A survey on wireless security: technical challenges, recent advances, and future trends. Proc. IEEE **104**, 1727–1765 (2016)
111. F. Restuccia, T. Melodia, Big data goes small: real-time spectrum-driven embedded wireless networking through deep learning in the RF loop, in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Paris (May 2019)

112. J. Jagannath, N. Polosky, A. Jagannath, F. Restuccia, T. Melodia, Machine learning for wireless communications in the Internet of things: a comprehensive survey. Preprint. arXiv:1901.07947
113. S. D'Oro, F. Restuccia, T. Melodia, Hiding data in plain sight: undetectable wireless communications through pseudo-noise asymmetric shift keying, in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Paris (May 2019)
114. F. Restuccia, S. D'Oro, T. Melodia, Securing the internet of things in the age of machine learning and software-defined networking. IEEE Internet Things J. **5**(6) (2018)
115. F. Restuccia, N. Ghosh, S. Bhattacharjee, S.K. Das, T. Melodia, Quality of information in mobile crowd sensing: survey and research challenges. ACM Trans. Sensor Netw. **13**(4), 34 (2017)

Chapter 9
Proactive User Authentication Using WiFi Signals in Dynamic Networks

Hongbo Liu, Yan Wang, Jian Liu, and Yingying Chen

Abstract User authentication is the critical first step of network security to detect identity-based attacks and prevent subsequent malicious attacks. However, the increasingly dynamic mobile environments make it harder to always apply the cryptographic-based methods for user authentication due to their infrastructural and key management overhead. Exploiting non-cryptographic-based techniques grounded on physical layer properties to perform user authentication appears promising. To ensure the security of mobile devices in dynamic networks, we explore to use fine-grained channel state information (CSI), which is available from off-the-shelf WiFi devices, to perform proactive user authentication. We propose a user-authentication framework that has the capability to proactively request CSI and build the user profile resilient to the presence of the spoofer. Our machine learning based user-authentication techniques can distinguish two users even when they possess similar signal fingerprints and detect the existence of the spoofer in dynamic network environments. Extensive experiments in both office and apartment environments show that our framework can remove the effect of signal outliers and achieve higher authentication accuracy compared to existing approaches that use received signal strength (RSS).

H. Liu
Indiana University Purdue University Indianapolis, Indianapolis, IN, USA
e-mail: hl45@iupui.edu

Y. Wang
Binghamton University, Binghamton, NY, USA
e-mail: yanwang@binghamton.edu

J. Liu · Y. Chen (✉)
Rutgers University, New Brunswick, NJ, USA
e-mail: jianliu@winlab.rutgers.edu; yingche@scarletmail.rutgers.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances
in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_9

9.1 Introduction

In recent decades, wireless technologies evolved rapidly, and people can access network services through the ubiquitous wireless networks almost at everywhere. However, because the wireless medium is open to any user, it is challenging to protect wireless networks from adversaries who can eavesdrop or intercept any wireless transmission [15]. For example, an adversary can passively monitor a wireless network to obtain valid device identities and further launch identity-based attacks. Such attacks could serve as a basis for launching a variety of malicious attacks across multiple network layers [5]. The adversary can easily perform such identity-based attacks in WiFi networks. For example, the adversary can spoof an Access Point (AP) and denial all the services (i.e., rogue AP attack) [29]. Although existing cryptographic-based authentication techniques (such as WiFi Protected Access and 802.11i) can detect the spoofed data frames, the 802.11 management frames are still open to attacks [20]. Furthermore, cryptographic-based authentication is becoming harder to deployed in dynamic mobile environments because of the requirement of infrastructural and key management overhead [1, 4, 6].

Recently, researchers have developed non-cryptographic-based authentication methods to complement and enhance the existing cryptography-based schemes [2, 5, 8]. For example, the RF channel based authentication schemes use the unique characteristics of the Received Signal Strength (RSS) of wireless packets [5] or the Channel Impulse Response (CIR) of a single frequency [21] to differentiate users. These schemes are developed based on the intuition that the wireless channel properties (i.e., RSS and CIR) present unique spatial patterns due to path loss and multi-path effects. Therefore, an adversary will incur different RSS or CIR patterns when he resides at a different location from the legitimate user, which could be utilized to differentiate the legitimate user from adversaries. However, the effectiveness of the user authentication methods is largely limited since the RSS and CIR extracted from a single frequency only provide coarse-grained information about the wireless channel. For instance, the RSS-based authentication can hardly differentiate two users who have similar RSS signatures even though they may be at the locations far away from each other [5].

Different from the existing work, we propose to leverage the fine-grained physical layer information carried by the orthogonal frequency-division multiplexing (OFDM) signals to perform user authentication. In particular, we find that the Channel State Information (CSI) derived from the channel response of multiple OFDM subcarriers [9] has unique spatial patterns with a fine granularity, which could facilitate accurate user authentication. Compared to the existing channel-based (i.e., RSS and CIR) approaches, our CSI-based approach can accurately discriminate the legitimate user from a spoofing attacker even though the signal fingerprints are similar. In addition, the detailed channel information is available at per packet level, which ensures much higher granularity than the existing channel-based approaches in both spatial and temporal domains. In this work, we associate each user with her wireless device, which is not accessible to other users. Thus, every user has a distinct CSI profile corresponding to her device and location. Thus, our user authentication method could examine the CSI from the associated wireless device and determine whether it is from a legitimate user or not.

Recently, Jiang et al. [13] devises an authentication system that utilizes a sliding-window based technique to construct CSI user profiles. The proposed system assumes that the CSI measurements collected for user profile construction are benign (i.e., without the presence of an identity-based attacker). However, such an assumption is not practical as the identity-based attacker could be present at any time. If the CSI measurements from the attacker are used to build the user profile, the system will misclass the user profile and falsely authenticate the attacker. To tackle the practical security issues, we develop effective approaches to construct the legitimate user profile considering the existence of a spoofer and perform robust user authentication under various adversarial scenarios, including the extreme case when only the attacker is active. Toward this end, we develop a framework that consists of two major components: *Attack-resilient Profile Builder* and *Profile Matching Authenticator*. The Attack-resilient Profile Builder is designed to accurately construct user profiles for legitimate users even when the identity-based attackers are present. The Profile Matching Authenticator is developed to perform robust per-packet user authentication based on real-time CSI measurements using machine-learning based technologies. Our framework is also among the first to consider the effect of different modulation and coding scheme rates of CSI and utilize the knowledge to achieve more accurate user authentication.

Our major contributions are summarized as follows: first, we show that CSI is feasible to perform robust user authentication even when different users have similar signal fingerprints and makes the fine-grained user authentication achievable in practice. Second, we develop a CSI-based user authentication framework that can effectively build user profiles under the presence of spoofing attacks and achieve higher authentication accuracy when comparing with the existing channel-based (e.g., RSS-based) methods. Last but not least, our CSI-based user authentication framework is validated by extensive experiments in both office and apartment environments using commodity WiFi devices. The experimental results demonstrate that our framework is highly robust and effective under various identity-based attacking scenarios without adding overhead to WiFi devices.

The rest of the chapter is organized as follows. In Sect. 9.2, we put our work in the context of the related studies. Section 9.3 describes the attack model and our framework overview, and Sect. 9.4 presents the feasibility of using CSI to perform user authentication. The proposed Attack-resilient Profile Builder based on clustering analysis is detailed in Sect. 9.5. Section 9.6 further discuss the Profile Matching Authenticator grounded on machine learning techniques. We present the details of our experimental setup and methodology in Sect. 9.7, which also include the performance evaluation of our CSI-based authentication framework in both office and apartment environments and different attacking scenarios. Finally, we conclude our work in Sect. 9.8.

9.2 Related Work

The traditional approach to provide user authentication is to use cryptographic-based authentication. For example, Wu et al. [27] have introduced a secure and efficient key management (SEKM) framework. SEKM builds a Public Key Infrastructure (PKI) by applying a secret sharing scheme and an underlying multicast server group. Wool [26] implements a key management mechanism with periodic key refresh and host revocation to prevent the compromise of authentication keys. The application of cryptographic authentication requires reliable key distribution, management, and maintenance mechanisms, which reduce its usability in a dynamic mobile wireless environment (i.e., lacks of a fixed key management infrastructure) or resource-constrained wireless networks (i.e., limited resources on wireless devices).

Recently non-cryptographic based authentication has drawn considerable attention [30]. In general, non-cryptographic solutions can be categorized into four groups: software based, hardware based, biometric and physical-trait based, and channel-signature based. Software based authentication basically relies on the unique characteristics of the software programs or protocols running on the devices [8, 24], whereas hardware based authentication leverages the unique hardware traits such as channel-invariant radiometric [2, 22] and clock skews [12, 17] to identify users. Biometric and physical-trait based authentication relies on the behavioral modalities including on-screen touch and finger movement patterns [7, 19]. And channel-signature based authentication schemes are proposed to use either Received Signal Strength (RSS) [5, 14, 28, 29, 31] or Channel Impulse Response (CIR) [21, 25] to identify users. The major advantage of using channel signatures is that it exploits the naturally available random and location-distinct characteristics of the wireless channel, which is very hard to falsify, for user authentication.

For the channel based user authentication using RSS, a series of approaches [5, 28, 29] have been proposed to detect identity-based attacks, determine the number of attackers when multiple adversaries masquerading as the same node identity, and localize the adversaries. Reciprocal Channel Variation-based Identification (RCVI) [31] exploits the reciprocity of RSS variance to decide if all packets come from a single or more than one sender. Ensemble [14] leverages a user's growing collection of trusted devices that analyze variations in RSS to determine whether the pairing devices are in physical proximity to each other. It is important to note that although RSS is available on the current wireless devices, RSS is known to be sensitive to the multipath effects and affected by the transmission power level. As a result, a legitimate user may be mistakenly regarded as the malicious user due to the inherent RSS variance. Different from RSS which is readily available in the existing wireless infrastructure, CIR is usually extracted from the specialized devices such as Field-Programmable Gate Array (FPGA) [25] and Universal Software Radio Peripheral (USRP) [21], which limit its practical usage in real-world scenarios.

Different from the previous work, we propose to use Channel State Information (CSI), a readily available fine-grained channel information from the current commercial hardware (i.e., 802.11 a/g/n devices), which represents both amplitude and phrase for each subcarrier on the 802.11 a/g/n OFDM system. Exploiting CSI has the potential to achieve much higher granularity (in both spatial and temporal) for user authentication than applying existing channel based (i.e., RSS and CIR) authentication methods. The most related work to us is CSITE [13], which utilizes CSI magnitude measurements averaged over time to generate profiles for legitimate users. They assume the CSI collected over time is benign and there is no identity-based spoofing attack present when building the profile. However, in practice the spoofing attack could present at any time. Thus, the profiles built under such attacks cannot represent legitimate users and may lead to authenticate malicious users falsely. In our work, we develop an Attack-resilient Profile Builder, which has the ability to detect the presence of spoofing attacks when building profiles for legitimate users. Furthermore, we study the effect of different modulation and coding scheme rates to CSI to achieve a higher accuracy of user authentication under both single antenna and multiple antenna cases.

9.3 Attack Model and System Overview

In this section we first introduce the attack model we consider in this work. We then present the flow of our proposed CSI-based user-authentication framework.

9.3.1 Attack Model

User authentication is a technique of confirming the identity of a user. Based on the user authentication result, a system can determine whether a user is allowed to access certain restricted services, such as restricted access of certain web sites and enterprise data retrieval [23]. User authentication is particularly challenging in wireless networks as it is very hard, if not impossible, to physically confirm the truth of a user's identity due to the open nature of the wireless medium. In our user-authentication framework, we focus on the identity-based attack, in which an adversary can collect a legitimate user's identity and then masquerades as the legitimate user to pass the user authentication process [5]. The identity-based attack is very harmful as once passing the user authentication, the adversary can gain certain access privileges and further launch a variety of malicious attacks. For example, an adversary could easily obtain the Media Access Control (MAC) address of a legitimate WiFi device by passively monitoring the wireless traffic and then impersonate as the legitimate device by changing its MAC address. Another example is that by masquerading as an authorized wireless access point (AP) or an authorized client, an attacker could launch a variety of attacks including session hijacking, denial-of-service (DoS) attacks, or falsely advertise services to wireless clients [29].

In this work, we consider the identity-based attack can be present at any time. That is, different from the previous work, which only considers the presence of such an attack during the authentication phase, we take the view point that the identity-based attacks could be present at any time in real-world scenarios even when building profiles for legitimate users. Once such an attack is present in the network, the adversary spoofs the legitimate user's device identity (e.g., WiFi device's MAC address) to send out packets. Once the attacker obtains the legitimate user's device identity, it can access the network with or without the presence of the legitimate user. Furthermore, the spoofer can be either static or mobile, whereas the legitimate device is mostly placed at a fixed position but could be moved from one location to another (e.g., the user walks from one office room to a meeting room). The movement of the device can be detected using the existing techniques [3, 16, 18] (e.g., examining the variance of the wireless signal). In addition, we assume the attacker does not have the capability to capture and replay the CSI, thus the attacker cannot alter or jam the signals.

9.3.2 System Overview

Our basic idea is to profile the user by exploiting the readily available fine-grained CSI extracted from orthogonal frequency-division multiplexing (OFDM) based wireless networks, such as 802.11 a/g/n networks. CSI reveals the wireless channel response depicting the amplitudes and phases of every OFDM subcarrier. In general, CSI measurements from each user present a unique pattern corresponding to the wireless communication channel. Such CSI patterns can be extracted and utilized to uniquely identify each user. If the observed wireless packet (from a wireless device identity) contains a different CSI pattern from the legitimate profile, the network will raise an alert indicating possible identity-based attack and fails the user authentication on the specific device identity.

Our proposed user authentication framework, as depicted in Fig. 9.1, consists of two main components: Attack-resilient Profile Builder and Profile Matching Authenticator. The network implementing this framework will keep monitoring the wireless traffic and examining CSI measurements from each packet based on the device's identity.

Attack-Resilient Profile Builder The novelty of our profile builder is that it has the capability to build the actual user profile under the presence of the spoofer when building the user profile. When building the user profile for a specific user identity (ID), the presence of the spoofer will cause the CSI measurements collected from this ID containing the mixture signal information from both the legitimate user and spoofer. As a result, the profile built under such a scenario is thus undermined by the spoofer, leading to mistakenly authenticate the spoofer or deny the legitimate

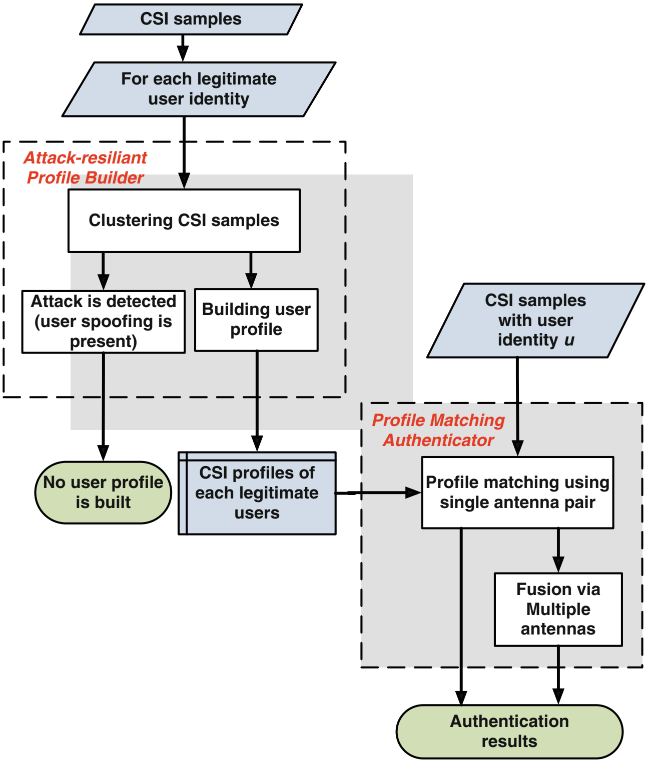

CSI samples
For each legitimate user identity
Attack-resiliant Profile Builder
Clustering CSI samples
Attack is detected (user spoofing is present)
Building user profile
No user profile is built
CSI profiles of each legitimate users
CSI samples with user identity *u*
Profile Matching Authenticator
Profile matching using single antenna pair
Fusion via Multiple antennas
Authentication results

Fig. 9.1 Overview of the CSI-based user authentication framework

user's access. Our profile builder employing clustering analysis can separate the CSI measurements (from the legitimate user) from the ones (from the spoofer) and determine the presence of the spoofer. It can thus ensure the legitimacy of the user profile construction.

Furthermore, when the legitimate user moves from current location to another, e.g., from office to a meeting room, our framework can adaptively rebuild the user's profile. This rebuilding procedure can be user triggered or triggered after detecting the user movement based on existing techniques [3, 16, 18].

Profile Matching Authenticator This component examines the real-time CSI measurements per packet from a device ID and performs user authentication by performing user profile matching. It is grounded on the machine-learning based techniques and raises an alert if the profile matching fails. Our authenticator aims to achieve fine-grained user authentication as it can work at per packet level—authenticating each packet of the device ID. It is capable to authenticate different users even when they possess similar signal fingerprints due to the complex environment setup in real-life. The authenticator works well under both single antenna as well as multiple-antennas cases (using data fusion).

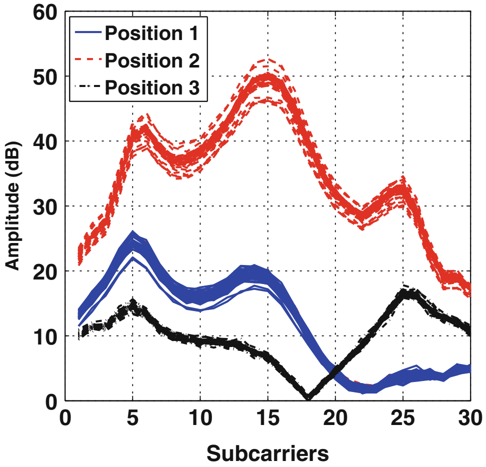
9.4 Feasibility Study of CSI-Based User Authentication

In this section, we first provide the background of CSI measured from OFDM subcarriers. We then discuss the feasibility of using CSI for user authentication. We next present our data pre-processing techniques applied to CSI measurements for more reliable user authentication.

9.4.1 Preliminary

Our authentication framework exploits the CSI measured from OFDM subcarriers, a reliable and fine-grained description of channel characteristics, for user authentication. OFDM technique has been extensively used in wireless communication systems to improve the communication performance by utilizing the frequency diversity of wireless channels. For example, OFDM is used in popular wireless networks including IEEE 802.11 a/g/n, WiMAX, 4*G* and Digital Subscriber Line (DSL). OFDM is a method of encoding data streams on multiple carrier frequencies. In particular, Data in OFDM is split into multiple streams, which are coded and modulated respectively into different subcarriers. The frequency of each subcarrier is designed to be orthogonal to each other, so that the interference during transmission is minimized. For example, for the OFDM employed by the 802.11 a/g/n physical layer, a relatively wideband channel (or carrier) with 20 or 40 MHz is partitioned into 54 or 108 subcarriers for data transmission, so that each subcarrier can be used as a narrowband channel. This inspires us to exploit the channel state information (CSI) extracted from OFDM subcarriers for user authentication, which can provide a finer granularity of the channel information and has the potential to achieve higher accuracy for user authentication in practice. Figure 9.2 depicts the

Fig. 9.2 Example for channel state information of OFDM that is collected at three different positions



amplitude of channel state information across 30 subcarrier groups at three different positions. For each position, the CSI of 50 packets are measured from an Intel WiFi 5300 card in a laptop [9].

9.4.2 Feasibility Study

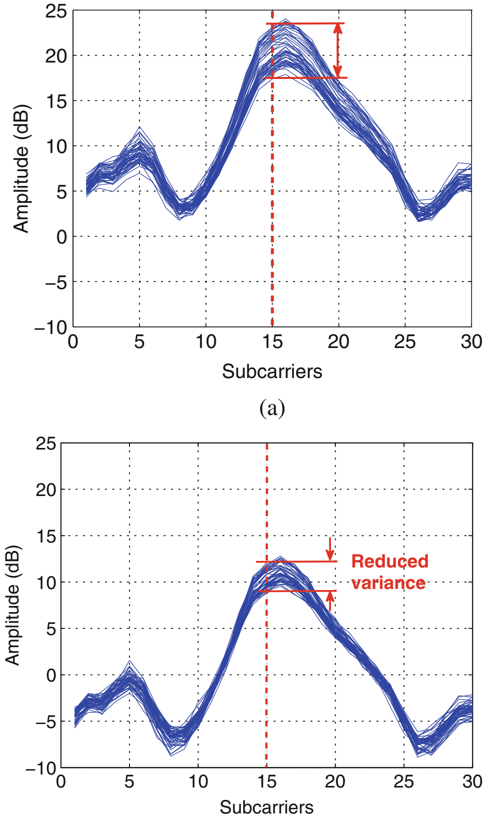
To be able to use CSI for user authentication, the measured CSI from different devices should satisfy the *uniqueness* requirement. That is, the CSI measured at different devices resided at different locations should be distinct, while the CSI collected from different packets emitted by the same device should be similar, if not identical. We observe in Fig. 9.2 that the amplitude of CSI at different subcarriers is different due to frequency diversity. Furthermore, the CSI shape from these three devices at different locations are distinct. This is because the CSI is the reflection of the complicated wireless channel and is affected by the wireless environment due to reflection, refraction, shadowing, etc. The CSI decorrelates with location rapidly. If two users are located at different locations, the profile of CSI should differ significantly. Additionally, we observe that the CSI of multiple packets from the same device at a fixed location exhibit the same trend, which indicates that an unique profile could be built for each user and serves as the basis for user authentication.

Note that compared to the RSS, which only provides overall received power for each packet, CSI provides fine-grained channel information, i.e., channel responses on multiple subcarriers. Therefore, instead of deploying multiple landmarks or monitors to collect multi-dimensional RSS reading for user authentication purpose, a single monitor can provide multi-dimensional channel state information sufficient for user authentication. Furthermore, since the widely adopted IEEE 802.11n standard [11] already defines a mechanism to exchange detailed CSI between a pair of wireless devices, employing CSI as an unique means for user authentication will not involve extra communication cost for the prevalent WiFi networks.

Data Preprocessing In our study, we observe that the mean amplitude value of CSI measurement may shift over time. We call such a mean value shift as *temporal bias*, and it will result in inaccurate CSI profile construction for user authentication. Therefore, our framework develops a data preprocessing strategy to cope with CSI samples to mitigate the effects caused by such temporal bias.

In particular, we observe a shift on the amplitude of a specific subcarrier due to the interference presented at either transmitter or receiver. Figure 9.3a plots the curve of the CSI sample in a packet and many curves are collected over time. It shows that the amplitude of each subcarrier in CSI samples varies over time. Our data preprocessing strategy adjusts the mean value of the CSI sample (from a specific packet) to zero. This helps to reduce the overall variance across the subcarriers on CSI measurements before performing user authentication. To illustrate, we denote the raw CSI sample per packet from a particular user u as a k-dimensional vector C_u, and the preprocessed CSI sample can be obtained as:

Fig. 9.3 CSI samples before and after data processing



(a)

(b)

$$C_u = C_u - 1_{1\times K}\frac{1}{K}\sum_{k=1}^{K} C_u(k), \tag{9.1}$$

where K is the number of subcarriers within a single CSI sample, and $1_{1\times K}$ is a K-length all-one vector. After applying the data preprocessing strategy, the updated CSI samples will have smaller variance and reduced amplitude bias on each subcarrier as shown in Fig. 9.3b. In addition, the wireless devices in the 802.11n network are usually equipped with multiple-antennas. Thus, the CSI samples collected from each channel between the transmitting antenna i and receiving antenna j of two communicating devices will go through the pre-process as shown in Eq. (9.1), where C_u will be replaced by $C_u^{i,j}$.

9.5 Attack-Resilient User Profile Builder

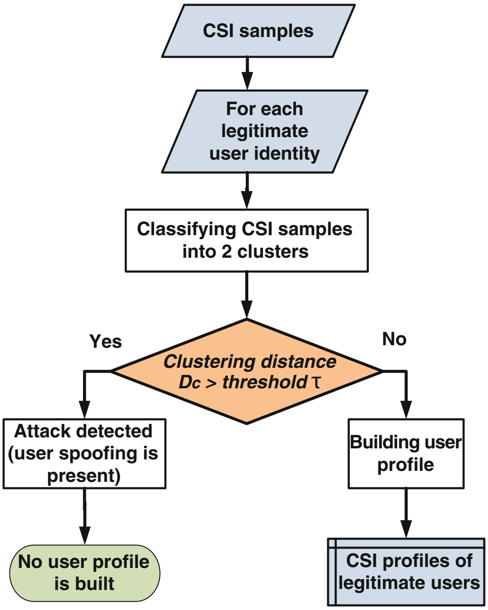
In this section, we describe the attack-resilient profile builder which employs clustering analysis on CSI measurements to determine whether the network environment is benign or the spoofer is present when constructing the user profile.

9.5.1 Basic Idea

Since the spoofing attack could be present at any time, we need to determine whether a spoofer is present when constructing a user profile. Our attack-resilient profile builder aims to ensure the legitimacy of the user profiles even under a malicious wireless environment. The relational behind our attack-resilient profile builder is that the CSI measurements of each device presents unique spatial characteristics: the CSI has strong spatial correlation with the device's location. Although the wireless channel may fluctuate over time, the CSI of wireless packets from one device at a fixed location should be clustered together in the multi-dimensional signal space constructed by CSI measurements. For example, the 30 subcarriers obtained in our experiments can form a 30-dimensional CSI space, and the amplitudes of the CSI from the 50 packets in Position 1 are clustered together (i.e., has a constant shape) in CSI space as shown in Fig. 9.2. Furthermore, the CSI measurements of the wireless packets collected from another device resided at a different location (Position 2) should form a different cluster in the CSI space. Thus, when the environment is benign, the CSI measurements from a particular device identity should be clustered together and form one cluster in the CSI space, while under the spoofing attack, the spoofer utilizes the same device identity as the legitimate user to transmit packets, and the CSI readings of the device identity are the mixture readings from both legitimate user and the spoofer, resulting in more than one CSI cluster.

To determine whether the network environment is benign, Our framework applies clustering analysis to partition the CSI from one device identity into two clusters. Under normal conditions without spoofing, the distance between the partitioned two CSI clusters should be small since there is basically only one cluster from a single device at a physical location. However, under a spoofing attack, there is more than one devices at different physical locations claiming the same device identity. As a result, more than one CSI clusters will be formed in the CSI space. Therefore, the distance between two partitioned clusters will be large as the cluster centers are derived from the different CSI clusters associated with different locations in physical space. Therefore, by examining the distance between the two partitioned CSI clusters, any presence of the spoofing attack can be determined when building user profiles. The flow of the Attack-resilient Profile Builder is shown in Fig. 9.4. Only when there is no spoofing attack present, the profile of the legitimate user will be built.

Fig. 9.4 Work flow for the attack-resilient profile builder

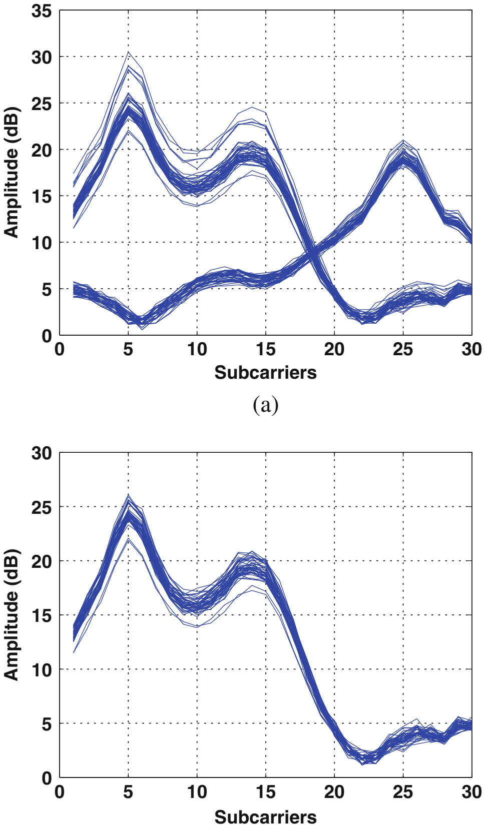

CSI samples
For each legitimate user identity
Classifying CSI samples into 2 clusters
Clustering distance Dc > threshold τ
Yes
No
Attack detected (user spoofing is present)
Building user profile
No user profile is built
CSI profiles of legitimate users

9.5.2 *Algorithm Description*

9.5.2.1 Modulation and Coding Scheme Study

WiFi devices usually use a fixed range of modulation and coding scheme (MCS) for data transmission. We find in our experiments that the modulation and coding scheme occasionally changes to a different one and then switches back due to the variation of the wireless channel. And the occasionally changed modulation and coding scheme creates outliers in the CSI measurements. Thus, our framework first performs outlier filtering based on the modulation and coding scheme used for packet transmission before conducting clustering analysis. In particular, MCS is a specification of the high-throughput (HT) physical layer (PHY) parameter in 802.11n standard [11]. It contains the information of the modulation order (e.g., BPSK, QPSK, 16-QAM, 64-QAM), the forward error correction (FEC) coding rate, etc. Each 802.11n packet header (at 2.4GHz band) contains a 16-bit MCS, which can be extracted together with the CSI sample of each packet.

Figure 9.5a shows the raw CSI measurements for a wireless device with two clusters formed in our experiments. Under such cases, the MCS rate is changing according to the channel condition, and we can observe CSI samples resulting from different MCS rates. For these cases we find the MCS values are greater than 263, different from most of the other testing cases in both the lab and apartment environments. We thus filter out CSI for the packets with MCS value greater than 263, which corresponds to single spatial stream with transmission rate 60 Mbps



(a)

(b)

Fig. 9.5 CSI samples before and after filtering based on MCS rate

[11]. After filtering out the outliers, the CSI coming from the rest of the packets exhibit the similar shape (i.e., form one cluster in the CSI space) as shown in Fig. 9.5b.

9.5.2.2 Clustering Analysis

We utilize the K-means algorithm to partition the filtered CSI measurements from the device identity u into two clusters. The K-means algorithm is one of the most popular iterative descent clustering methods [10]. The squared Euclidean distance is chosen as the dissimilarity measure. If there are S CSI samples from the device u, the K-means clustering algorithm partitions S CSI samples into K disjoint subsets L_k containing S_k sample points so as to minimize the sum-of-squares criterion:

$$J_{min} = \sum_{k=1}^{K} \sum_{C_{u,s} \in L_k} \|C_{u,s} - \boldsymbol{\mu_k}\|^2, \qquad (9.2)$$

where $C_{u,s}$ is a CSI vector representing the CSI value for the sth packet and $\boldsymbol{\mu_k}$ is the geometric centroid of the sample points for L_k in CSI space. In our cluster analysis, we choose $K = 2$. We further choose the distance between two centroids as the test statistic $\mathbf{T}$ for identity-based attack detection,

$$D_c = ||\boldsymbol{\mu_k} - \boldsymbol{\mu_{k'}}|| \tag{9.3}$$

with $k, k^{'} \in \{1, 2\}$.

Under normal conditions in a benign network environment, the distance between the centroids from the K-means cluster analysis in CSI space should be close to each other, because there is only one cluster from a single device u at a physical location. However, when a spoofer is present, there is more than one devices residing at different physical locations, claiming the same device identity. The distance between two partitioned CSI clusters thus will be large. Through the analysis above, we show that the clustering method has the capability of detecting the presence of the spoofer by applying the threshold τ to the D_c as following:

$$\begin{cases} D_c > \tau \text{ attacker is present;} \\ D_c \leq \tau \text{ normal condition.} \end{cases} \tag{9.4}$$

9.5.2.3 User Profile Building

If the CSI samples are collected in a benign environment, the framework deposits the pre-processed CSI samples, C_u, as the profile for user u for future profile matching based authentication. We note that the user profile only requires a small number of packets, i.e. less than 100 packets.

If the user moves from one location to another (e.g., walks from his office to a meeting room), the user authentication framework will adaptively rebuild the user's profile. Following are possible two ways to update the user's profile: (1) the user can actively trigger the profile updating after he moves to a new place; (2) the profile updating can be triggered by detecting the user movement using existing techniques operating on wireless signals [3, 16, 18].

9.6 User Authentication Leveraging Profile Matching

In this section, we present our profile matching authenticator which uses machine-learning based methods for packet-level user authentication.

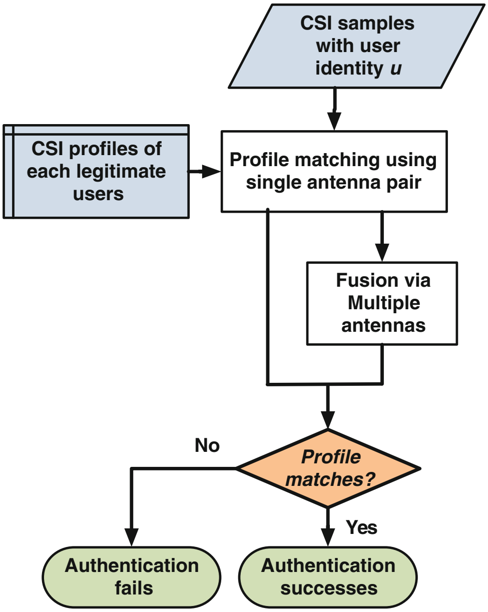

CSI samples with user identity *u*
CSI profiles of each legitimate users
Profile matching using single antenna pair
Fusion via Multiple antennas
No
Profile matches?
Yes
Authentication fails
Authentication successes

Fig. 9.6 Work flow for the profile matching authenticator

9.6.1 Basic Idea

The basic idea of our profile matching authenticator is using machine learning to determine whether the CSI measurement for the incoming packet with the user identity u matches the profile constructed at the profile builder. If the incoming CSI sample matches the user profile, the corresponding packet can be authenticated successfully as from the user u. Otherwise, the user authentication fails. Figure 9.6 illustrates the work flow of our profile matching authenticator. In particular, the profile matching scheme works at the packet level, which minimizes the latency of the authentication process. In addition, the packet-level authentication can also be used to monitor and count the number of packets injected by the attacker.

9.6.2 Approach Description

We next present the profile matching method using the CSI samples from a single antenna. We then present the profile matching using CSI samples from multiple antennas to improve the performance of user authentication.

9.6.2.1 Profile Matching Using a Single Antenna Pair

We perform the profile matching via the support vector machine (SVM) technique, which is a computationally efficient way of learning good separating hyperplanes in a high dimensional feature space. The CSI samples are used as features in the SVM to perform profile matching for each user. We first study the case using a single antenna pair for profile matching.

In this study, we consider the profile matching as a two-class pattern classification problem. The CSI sample C_u with user identity u denotes the data to be classified, where $u = 1, \cdots, U$ (with U as total number of legitimate users), and let scalar y denote its class ($y \in \{-1, 1\}$). We use $\{(C_{u,s}, y_{u,s}), s = 1, \ldots, S\}$ to denote a set of CSI samples associated with the user identity u. The challenge is how to construct a decision function $f(C_u)$ that correctly classifies the input CSI data, which could be different from all the constructed profiles.

If the constructed CSI user profiles are linearly separable, we can represent them with a linear function in the following form:

$$f(C_u) = w^T C_u + b \tag{9.5}$$

such that $f(C_{u,s}) \geq 0$ for $y_{u,s} = 1$ and $f(C_{u,s}) \leq 0$ for $y_{u,s} = -1$, where w and b represent the hyperplane $f(C_u) = 0$ separating two classes.

We seek to find such a hyperplane that maximizes the separating margins between the two classes. In particular, this hyperplane can be found by minimizing the following cost function:

$$\min J(w, \xi) = \frac{1}{2}\|w\|^2 + \Gamma \sum_{s=1}^{S} \xi_{u,s} \tag{9.6}$$

subject to the following constraints:

$$y_{u,s}(w^T \Phi(C_{u,s}) + b) \geq 1 - \xi_{u,s}, \xi_{u,s} \geq 0, s = 1, \cdots, S, \tag{9.7}$$

where $\Phi(\cdot)$ is a non linear operator mapping the CSI profile C_u to a higher dimensional space, Γ indicates the significance of the constraint violations with respect to the distance between the points and the hyperplane and ξ is a slack variable vector.

The mapping between the input CSI samples $C_{u,s'}$ and user profile $C_{u,s}$ is constructed in the form of the kernel function $Kernel(\cdot, \cdot)$, such as $Kernel(C_{u,s}, C_{u,s'}) = \Phi^T(C_{u,s})\Phi(C_{u,s'})$. Particularly, we choose a polynomial kernel as the mapping function and the problem in Eq. (9.6) can be expressed as:

$$\max_{\alpha_s}\{\sum_{s=1}^{S} \alpha_s - \frac{1}{2} \sum_{s=1}^{S} \sum_{s'=1}^{l} \alpha_u (y_{u,s} y_{u,s'} Kernel(C_{u,s}, C_{u,s'}))\alpha_{s'}\} \tag{9.8}$$

subject to the constraints:

$$\alpha_s \geq 0, \sum_{s=1}^{S} \alpha_s y_{u,s} = 0, \tag{9.9}$$

where α_s is the Lagrange multipliers associated with Eq. (9.7). Thus, the profile matching classifier for input CSI sample $C_{u,s'}$ is derived as:

$$f(C_{u,s'}) = sign(\sum_{s=1}^{S} (\alpha_s y_{u,s} Kernel(C_{u,s'}, C_{u,s}) + b)). \tag{9.10}$$

And the authentication result is determined as:

$$f(C_{u,s'}) = \begin{cases} 1 & success \\ -1 & failure. \end{cases} \tag{9.11}$$

9.6.2.2 Fusion via Multiple Antennas Pairs

When multiple antennas are available, we can further improve the performance of the user authentication accuracy. For example, we can employ a simple majority voting process to combine the independent profile matching results from different antenna pairs. Assume that the input CSI samples with user identity u between the transmitting antenna i and receiving antenna j are represented as $C_{u,s'}^{i,j}$, all the independent results from different antenna pairs consist of the voting set $\Omega = \{f(C_{u,s'}^{i,j}), 1 \leq i \leq I, 1 \leq j \leq J\}$, where I and J are the numbers of transmitting and receiving antennas respectively. Finally, the authentication result is given by:

$$f'(C_{u,s'}) = sign(\sum_{i=1}^{I} \sum_{j=1}^{J} f(C_{u,s'}^{i,j})). \tag{9.12}$$

If $f'(C_{u,s'}) = 1$, the authentication successes; otherwise it fails.

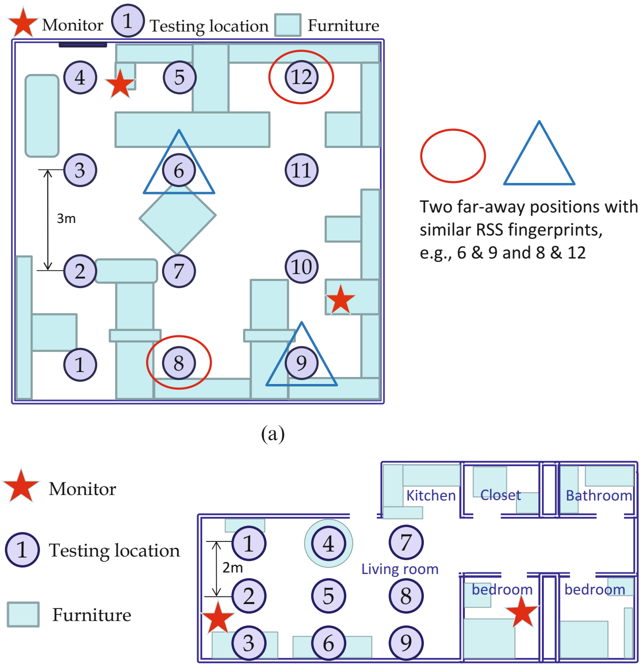
9.7 Performance Evaluation

In this section, we present the performance evaluation of the proposed CSI-based user authentication framework in two types of real environments, laboratory and apartment. We show that the CSI-based authentication framework is resilient to attacks, and outperforms existing RSS-based authentication methods.

9.7.1 Experimental Setup

We conduct experiments in a 802.11n WiFi network with two laptops (i.e., Lenovo T500 and T61) serving as monitors that collect the wireless packets. These two laptops run Ubuntu 10.04 LTS with the 2.6.36 kernel and are equipped with Intel WiFi Link 5300 wireless card. Both Intel wireless cards' drivers we installed are able to collect CSI information from frames transmitted in HT rate [11]. A commercial wireless access point, Linksys *E*2500, is sending out packets that can be captured by these two monitors. We use the *ping* command on two laptops to simulate the authentication packets continuously transmitted over the network. The packet rate is set to 10 packets/second. For each packet, we extract CSI for 30 subcarrier groups, which are evenly distributed in the 56 subcarriers of a 20 MHz channel [9]. We also record the RSS value of each packet for comparison.

We conduct experiments in two indoor environments, i.e., a laboratory and an apartment. The laboratory represents the typical office environment, which has office cubicle and many furniture that create complex multipath effects in a large room. The apartment, on the other hand, represents the typical home environment with small rooms and simple furniture. The size of these two environments are 11m × 12m and 11m × 6m, respectively. The experimental setups in these two environments are shown in Fig. 9.7. The numbered circles in the figures are the



Two far-away positions with similar RSS fingerprints, e.g., 6 & 9 and 8 & 12

(a)

(b)

Fig. 9.7 Experimental setups in (**a**) laboratory and (**b**) apartment

positions used to collect CSI data for evaluating our user authentication framework, and the two red stars represent two network monitors.

9.7.2 Experimental Methodology

In the experiments, we collect 400 packets at each location, and both CSI and RSS values of each packet are recorded. When using RSS measurements for user authentication, we employ the RSS values collected from two network monitors as the two-dimensional feature vector for clustering and profile matching, while our proposed CSI-based authentication framework only uses the CSI measurements from one network monitor to perform user authentication.

To evaluate the performance of our proposed framework, we examine two main attacking scenarios: (1) In the first attacking scenario, both the legitimate user and the attacker are present at the same time in the network. (2) In the second attacking scenario, after the attacker obtains the legitimate user's identity, only the attacker is active in the network. In order to obtain the statistical results, we choose all possible point pairs in both experimental environments and treat one point as the position of the legitimate user and the other point as the position of the attacker. We run the proposed framework through all possible combinations of point pairs. There are a total of 66 pairs for laboratory environment and 36 pairs for apartment environment. The experimental results are presented in the following sections for the attack resilient profile builder and profile matching authenticator.

9.7.3 Metrics

In order to evaluate the performance of our proposed user authentication framework, we define the following two metrics, attack detection ratio and authentication accuracy.

Attack Detection Ratio (During Profile Building) We define the attack detection ratio $\bar{R}$ as the number of correctly detecting the presence of spoofing attacks over the total number of experiments with spoofing attacks. The spoofing attacks presented when building the user profile belong to the attacking scenario 1. Given a total number of P attacking cases the attack detection ratio can be written as:

$$\begin{aligned} \bar{R} &= \frac{1}{P}\sum_{p=1}^{P} H_p \\ s.t.\ \ H_p &= \begin{cases} 0 & D_c \le \tau \\ 1 & D_c > \tau, \end{cases} \end{aligned} \tag{9.13}$$

where D_c is the distance between two centroids of clusters formed in the profile builder, and τ is the threshold used for spoofing attack detection.

Authentication Accuracy (During User Authentication) We define the authentication accuracy A_p as the number of correctly classified packets over the total number of packets collected in the pth attacking run. The attacks could belong to either the attacking scenario 1 or 2. We use $N_{u,p}$ to denote the number of packets that are sent by a legitimate user u and are correctly determined as from the user u by our system. Similarly, we use $N'_{u,p}$ to denote the number of packets sent by the adversary using the identity of the legitimate user u and are correctly determined as not from the user u. We then define the authentication accuracy for the pth experimental run as:

$$A_p = \frac{N_{u,p} + N'_{u,p}}{N_{a,p}}, \tag{9.14}$$

where $N_{a,p}$ is the total number of packets received with user identity u, and $N_{u,p} + N'_{u,p} \leq N_{a,p}$.

We further define the *average authentication accuracy* and *worst authentication accuracy* as shown below to evaluate the general and worst-case performance.

- **Average authentication accuracy:** Given P testing cases, the average authentication accuracy is given as:

$$A_{avg} = \frac{1}{P}\sum_{p=1}^{P} A_p. \tag{9.15}$$

- **Worst authentication accuracy:** The worst authentication accuracy chooses A_p from the attacking case with the smallest number of $N_{u,p}$ and $N'_{u,p}$:

$$A_{worst} = \min_{p} A_p. \tag{9.16}$$

9.7.4 Evaluation Results

9.7.4.1 Attack Detection Study During Profile Building

We first compare the effectiveness of our Attack-resilient Profile Builder when determining the presence of a spoofer (during profile building) using CSI to that using RSS. We examine the attack detection ratio by varying the threshold τ. As shown in Fig. 9.8, the results show that the averaged detection ratio for the proposed CSI based approach achieves 0.92 with the optimal distance threshold 17 dB in Fig. 9.8a, while the maximum detection ratio for the RSS-based method is only 0.4 with distance threshold 2 dB as shown in Fig. 9.8b. This observation indicates that our

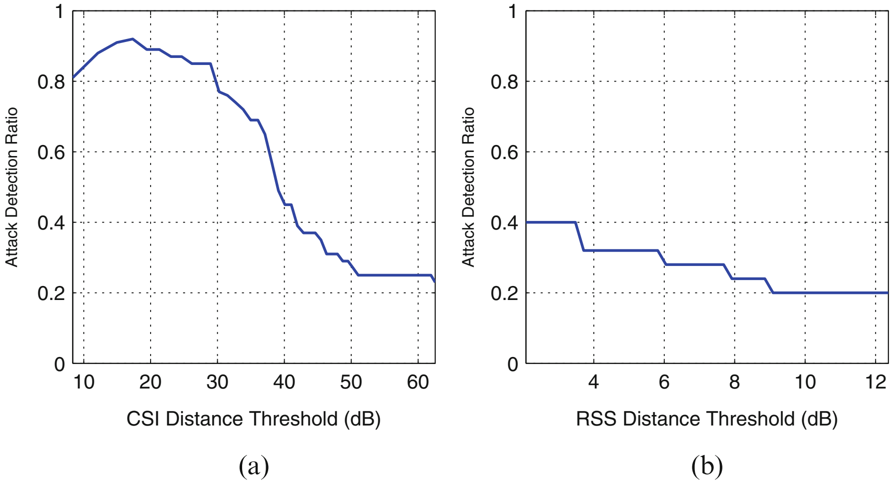


Fig. 9.8 Attack-resilient profile builder: attack detect ratio versus cluster distance threshold when a spoofer is present. (**a**) CSI-based. (**b**) RSS-based

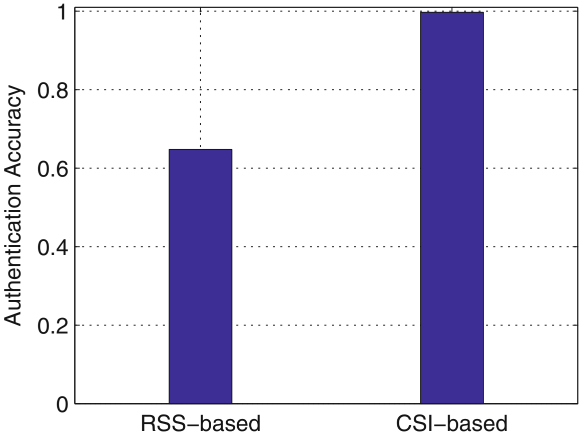
profile builder can effectively determine whether the network environment is benign or a spoofer is present when building the user profiles.

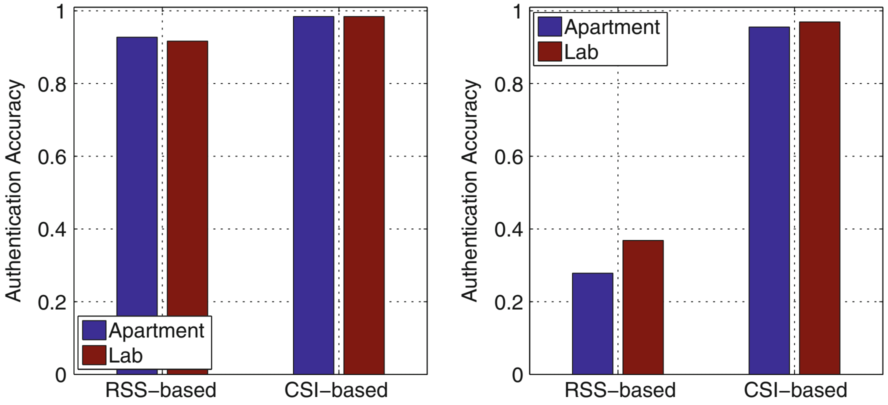
9.7.4.2 Authentication Accuracy Study

Discriminating Two Far-Away Users with Similar RSS Fingerprints Due to the irregularity of wireless signal propagation, two geographical distant users may share similar RSS signatures. For example, in Fig. 9.7a, two positions 6 and 9 are about 6-7 m away from each other, but their RSS fingerprints obtained from our same network monitor look similar; positions 8 and 12 present the same signal phenomenon. This makes RSS-based user authentication schemes suffer poor performance when two legitimate users (but physically separated) present the similar signal fingerprints. In particular, we observe that the authentication accuracy for RSS-based method degrades to only around 0.64 as shown in Fig. 9.9. However, our proposed CSI-based method could still achieve the authentication accuracy close to 1. The results confirm that CSI measurements provide fine-grained information on differentiating users, even when their RSS measurements are similar.

Comparison with RSS-Based Method We next study the overall performance of our CSI based user authentication method. Figure 9.10 shows the comparison of the authentication accuracy when using CSI-based and RSS-based methods in two different environments (i.e., a laboratory and an apartment). We note that the RSS-based method relies on RSS values collected from two network monitors to perform user authentication, while our proposed CSI-based authentication frame-

Fig. 9.9 Performance of the profile matching authenticator: authentication accuracy when two users possess similar RSS fingerprints





(a) (b)

Fig. 9.10 User authentication accuracy comparison between CSI-based and RSS-based methods. (**a**) A_{avg}. (**b**) A_{worst}

work only uses the CSI measurement from one antenna at one network monitor. We observe that our proposed CSI-based method outperforms the RSS-based method in both experimental environments. Specifically, Fig. 9.10a shows that the average authentication accuracy for CSI-based method is very high (above 0.984), and the RSS-based method has a lower authentication accuracy (i.e., 0.92). Furthermore, we show that the worst authentication accuracy for RSS-based method reduces to around 0.27 and 0.36 in the apartment and laboratory environments respectively, whereas our CSI-based method maintains the high authentication accuracy over 0.95 as presented in Fig. 9.10b. These observations strongly indicate the robustness of our CSI-based user authentication framework even when only a single antenna is used on WiFi devices.

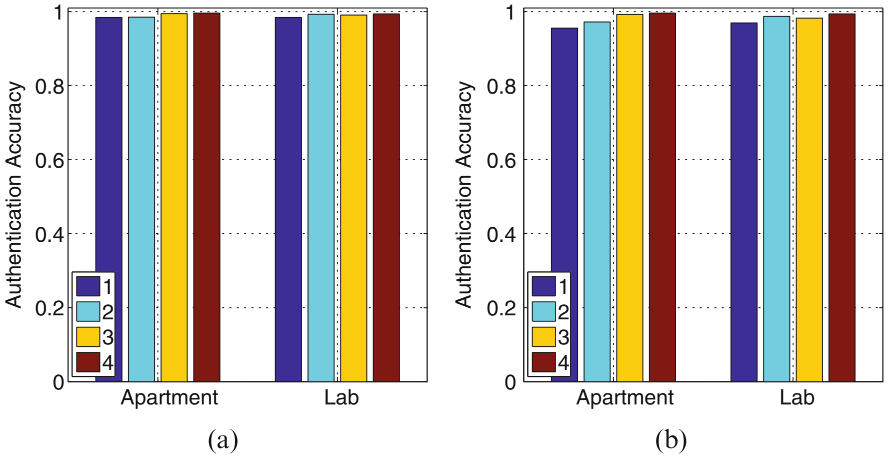
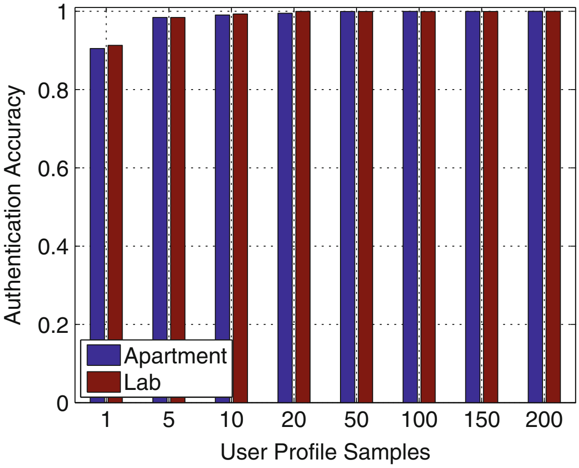


Fig. 9.11 CSI-based user authentication accuracy when involving single and multiple antennas. (**a**) A_{avg}. (**b**) A_{worst}

Impact from Single/Multiple Antennas We further examine the performance when employing measurements from multiple antennas. We expect that using measurements from multiple antennas can provide better reliability for user authentication. Figure 9.11 shows that both the average and worst authentication accuracy exhibit an increasing trend when more antennas are used. In particular, the authentication accuracy of using single antenna in the apartment and laboratory environments is over 0.95. When the number of antenna pairs (i.e., a set of transmitting and receiving antennas) increases from 1 to 4, the average authentication accuracy in laboratory and apartment further improves, and the worst authentication accuracy improves even more. We also observe that when using 3 antenna pairs in the laboratory environment the authentication accuracy has a slightly drop when comparing to that of using 2 antenna pairs. This is because although current commodity wireless devices are usually equipped with multiple antennas, the main antennas usually have better quality of signal reception. Therefore, including the CSI samples from the main antennas (i.e., using 1 or 2 antenna pairs in our experiments) results in better stability of user authentication.

Impact from User Profile Size Finally, we study how the number of packets (i.e., user profile size) employed to build the user profile affects the performance of our framework. We vary the size of user profile from 1 sample to 200 samples, and the corresponding average authentication accuracy is shown in Fig. 9.12. When the size of user profile increases, the authentication accuracy increases and then maintains at a high level (i.e., over 0.95). We note that even if the profile of each user contains only 1 CSI sample, the authentication accuracy is still over 0.91. These results demonstrate that our profile builder is highly effective in our CSI-based user authentication framework.

Fig. 9.12 Impact of user profile size on CSI-based user authentication accuracy



9.8 Conclusion

In this work, we explore the feasibility of utilizing channel state information (CSI) to perform practical user authentication in wireless networks. To achieve accurate user authentication, we propose a user authentication framework leveraging the fine-grained channel information revealed in CSI. The proposed framework consists of two major components, Attack-resilient User Profile Builder and Profile Matching Authenticator. Specifically, the Attack-resilient Profile Builder builds the profile for the legitimate user based on clustering analysis, and in the meanwhile it can intelligently determine whether the network environment is benign without the presence of the identity-based attack. The Profile Matching Authenticator performs packet level user authentication grounded on Support Vector Machine (SVM), and it is also capable to distinguish two users even when they possess the similar signal fingerprints. The extensive experimental results in both laboratory and apartment environments demonstrate that the proposed CSI-based approach is highly effective in contrast to the methods directly applying received signal strength.

Acknowledgements This work was supported in part by the National Science Foundation under grant numbers CNS-1318751, CNS-1318748 and Army Research Office W911NF-13-1-0288.

References

1. B. Azimi-Sadjadi, A. Kiayias, A. Mercado, B. Yener, Robust key generation from signal envelopes in wireless networks, in *Proceedings of the 14th ACM conference on Computer and communications security* (2007), pp. 401–410
2. V. Brik, S. Banerjee, M. Gruteser, S. Oh, Wireless device identification with radiometric signatures, in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking* (2008), pp. 116–127

3. G. Chandrasekaran, M.A. Ergin, M. Gruteser, R.P. Martin, J. Yang, Y. Chen, Decode: exploiting shadow fading to detect comoving wireless devices. IEEE Trans. Mob. Comput. **8**(12), 1663–1675 (2009)
4. O. Cheikhrouhou, A. Koubaa, M. Boujelben, M. Abid, A lightweight user authentication scheme for wireless sensor networks, in *IEEE/ACS International Conference on Computer Systems and Applications (AICCSA)* (2010), pp. 1–7
5. Y. Chen, J. Yang, W. Trappe, R.P. Martin, Detecting and localizing identity-based attacks in wireless and sensor networks. IEEE Trans. Veh. Technol. **59**(5), 2418–2434 (2010)
6. O. Delgado-Mohatar, A. Fúster-Sabater, J.M. Sierra, A light-weight authentication scheme for wireless sensor networks. Ad Hoc Netw. **9**(5), 727–735 (2011)
7. S. Govindarajan, P. Gasti, K.S. Balagani, Secure privacy-preserving protocols for outsourcing continuous authentication of smartphone users with touch data. IEEE Trans. Inf. Forensics Secur. **8**(1), 136–148 (2013)
8. F. Guo, T.-C. Chiueh, Sequence number-based mac address spoof detection, in *Recent Advances in Intrusion Detection* (2006), pp. 309–329
9. D. Halperin, W. Hu, A. Sheth, D. Wetherall, Predictable 802.11 packet delivery from wireless channel measurements. ACM SIGCOMM Comput. Commun. Rev. **40**, 159–170 (2010)
10. T. Hastie, R. Tibshirani, J. Friedman, *The Elements of Statistical Learning, Data Mining Inference, and Prediction* (Springer, New York, 2001)
11. IEEE Std. 802.11n-2009, Enhancements for higher throughput (2009). Available at http://www.ieee802.org
12. S. Jana, S.K. Kasera, On fast and accurate detection of unauthorized wireless access points using clock skews. IEEE Trans. Mob. Comput. **9**(3), 449–462 (2010)
13. Z. Jiang, J. Zhao, X.-Y. Li, J. Han, W. Xi, Rejecting the attack: source authentication for wi-fi management frames using csi information, in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2013)
14. A. Kalamandeen, A. Scannell, E. de Lara, A. Sheth, A. LaMarca, Ensemble: cooperative proximity-based authentication, in *Proceedings of the 8th International Conference on Mobile Systems, Applications, and Services* (2010), pp. 331–344
15. T. Karygiannis, L. Owens, Wireless network security. NIST Spec. Publ. **800**, 48 (2002)
16. K. Kleisouris, B. Firner, R. Howard, Y. Zhang, R.P. Martin, Detecting intra-room mobility with signal strength descriptors, in *The ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)* (2010), pp. 71–80
17. T. Kohno, A. Broido, K.C. Claffy, Remote physical device fingerprinting. IEEE Trans. Dependable Secure Comput. **2**(2), 93–108 (2005)
18. J. Krumm, E. Horvitz, Locadio: inferring motion and location from wi-fi signal strengths, in *MobiQuitous* (2004), pp. 4–13
19. L. Li, X. Zhao, G. Xue, Unobservable re-authentication for smartphones, in *Proceedings of the Network and Distributed System Security Symposium (NDSS)* (2013)
20. S. Mathur, W. Trappe, N. Mandayam, C. Ye, A. Reznik, Radio-telepathy: extracting a secret key from an unauthenticated wireless channel, in *Proceedings of the 14th ACM International Conference on Mobile Computing and Networking* (2008), pp. 128–139
21. S. Mathur, R. Miller, A. Varshavsky, W. Trappe, N. Mandayam, Proximate: proximity-based secure pairing using ambient wireless signals, in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services* (2011), pp. 211–224
22. N.T. Nguyen, G. Zheng, Z. Han, R. Zheng, Device fingerprinting to enhance wireless security using nonparametric bayesian method, in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2011), pp. 1404–1412
23. L. O'Gorman, Comparing passwords, tokens, and biometrics for user authentication. Proc. IEEE **91**(12), 2021–2040 (2003)
24. J. Pang, B. Greenstein, R. Gummadi, S. Seshan, D. Wetherall, 802.11 user fingerprinting, in *Proceedings of the 13th annual ACM International Conference on Mobile Computing and Networking* (2007), pp. 99–110

25. D. Shan, K. Zeng, W. Xiang, P. Richardson, Y. Dong, Phy-cram: physical layer challenge-response authentication mechanism for wireless networks. IEEE J. Sel. Areas Commun. **31**(9), 1817–1827 (2013)
26. A. Wool, Lightweight key management for ieee 802.11 wireless lans with key refresh and host revocation. ACM/Springer Wirel. Netw. **11**(6), 677–686 (2005)
27. B. Wu, J. Wu, E. Fernandez, S. Magliveras, Secure and efficient key management in mobile ad hoc networks, in *Proceedings of the 19th IEEE International Parallel and Distributed Processing Symposium (IPDPS)* (2005)
28. J. Yang, Y. Chen, W. Trappe, Detecting spoofing attacks in mobile wireless environments, in *6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks* (2009), pp. 1–9
29. J. Yang, Y. Chen, W. Trappe, J. Cheng, Detection and localization of multiple spoofing attackers in wireless networks. IEEE Trans. Parallel Distrib. Syst. **24**(1), 44–58 (2013)
30. K. Zeng, K. Govindan, P. Mohapatra, Non-cryptographic authentication and identification in wireless networks. Wirel. Commun. **17**(5), 56–62 (2010)
31. K. Zeng, K. Govindan, D. Wu, P. Mohapatra, Identity-based attack detection in mobile wireless networks, in *Proceedings of the IEEE International Conference on Computer Communications (INFOCOM)* (2011), pp. 1880–1888

Chapter 10
Multi-Carrier Jamming Mitigation: A Proactive Game Theoretic Approach

Koorosh Firouzbakht, Guevara Noubir, and Masoud Salehi

Abstract Wireless communications systems are highly vulnerable to jamming. There is a large body of research demonstrating the potential of smart-jamming that targets specific mechanisms in a wireless stack. Many proactive mitigation techniques sacrifice performance for guaranteed robustness. In this work, we develop a proactive mitigation approach for multi-carrier wireless links. The approach is formulated within the framework of game theory. We show that the interaction between a multi-carrier multi-rate system (in particular OFDM) and a power-limited jammer can be formulated as a constrained zero-sum or a bimatrix game. We show that the Nash equilibrium strategies can be derived analytically and numerically and we apply them to the special case of IEEE 802.11 OFDM links.

10.1 Introduction

The many desirable characteristics of the *Orthogonal Frequency Division Multiplexing* systems (OFDM) such as high spectral efficiency, high data rates, robustness in multipath fading channels and ease of implementation have made OFDM the primary physical layer solution for most modern wireless communication systems. *Wireless Local Area Networks* (WLAN) based on different flavors of the IEEE 802.11 or *Wireless Metropolitan Area Networks* (WMAN) based on IEEE 802.16 both use OFDM as the main physical layer modulation scheme. Additionally, most leading cellular technologies such as 4G LTE standard rely on OFDM at the physical layer.

Nevertheless, it has been shown in the literature that current implementations of OFDM are vulnerable to a variety of jamming attacks specially due to OFDM sensitivity to channel estimation and synchronization between the transmitter and

K. Firouzbakht · G. Noubir (✉) · M. Salehi
Northeastern University, Boston, MA, USA
e-mail: firouzbakht.k@husky.neu.edu; g.noubir@northeastern.edu; noubir@ccs.neu.edu; m.salehi@northeastern.edu

C. Wang and Z. Lu (eds.), *Proactive and Dynamic Network Defense*, Advances in Information Security 74, https://doi.org/10.1007/978-3-030-10597-6_10

the receiver [1]. For instance, Clancy and Goergen [2] studies the possibility of jamming the channel estimation procedure as an efficient type of attack. This work suggests that targeting the accuracy of channel state information estimation requires significantly less power than jamming the whole frequency band.

Other work studied jamming attacks that prevents the receiver from ever acquiring proper synchronization [3]. To achieve this goal, the adversary targets the symbol timing estimation algorithm, the first step in the synchronization process. This work suggests that a jammer who exploits the weakness in the timing estimation algorithm can cause massive errors to all synchronization estimates.

Other work focused on targeting the coding and interleaving scheme [4]. The authors discovered that in the case of IEEE 802.11ag, the coded bits are interleaved in a deterministic patterns over the OFDM sub-carriers. They demonstrated that it is possible for an adversary to block Wi-Fi packets at a cost 2–3 orders of magnitude lower than full-band jamming.

Game theory has also been used to study jamming games in OFDM systems, for instance, reference [5] considers jamming in a wireless OFDM network with transmission costs for both jammer and transmitter. This work uses the general-sum framework to model the jamming problem. The numerical example in this work suggests that when the jammer is close to the base station, the jammer should pay less attention to the subchannels with poor quality and spend more energy on the subchannels with good quality which in turn, forces the transmitter to use the resources of the bad quality subchannels.

In this chapter, we study the performance of an adaptive OFDM wireless communication system under power limited jamming using game theoretic approaches. We show that with modest assumptions, this problem can be formulated into either the constrained zero-sum, or the constrained bimatrix games. We presents some of fundamental of these framework here, for a detailed study of these frameworks we refer the reader to [6] and [7].

10.2 Communications and Adversary Models

In this section, we briefly introduce our system model and discuss the motivation behind our work. The details of our model will be discussed in the sections that follows. The transmitter and the receiver are communicating over a wireless noisy channel that is subjected to an adaptive adversary. The communicating nodes are using an *adaptive Orthogonal Frequency Division Multiplexing* (adaptive OFDM) to communicate. The transmitter adaptively changes the subcarriers' data rates such that the overall throughout of the wireless link is maximized.

On the other hand, the jammer, also adaptively, jams the OFDM subchannels with different jamming powers, in order to degrade the performance of the wireless link. We assume that the jammer can use arbitrary jamming powers and can jam any subchannel that he wishes but for practical reasons, he must maintain a maximum jamming power and energy. Our goal is to model this jamming problem and study the long term achievable performance of this adaptive OFDM system.

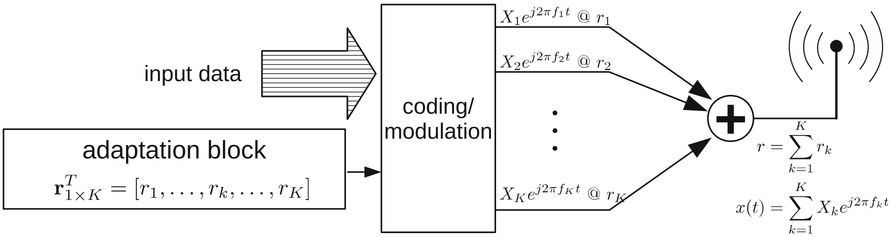

input data
adaptation block
$\mathbf{r}^T_{1\times K} = [r_1, \ldots, r_k, \ldots, r_K]$
coding/ modulation
$X_1 e^{j2\pi f_1 t}$ @ r_1
$X_2 e^{j2\pi f_2 t}$ @ r_2
$X_K e^{j2\pi f_K t}$ @ r_K
$r = \sum_{k=1}^{K} r_k$
$x(t) = \sum_{k=1}^{K} X_k e^{j2\pi f_k t}$

Fig. 10.1 The transmitter model

10.2.1 Communications Model Under Jamming

Consider an OFDM wireless communication system with K subchannels where the bandwidth of each subchannel is Δf. The transmitter has an adaptation block which enables him to jointly change/adapt his channel coding rate and modulation scheme for each subcarrier (see Fig. 10.1). For convenience, we assume the transmitter uses time domain channel coding and subchannel bandwidth is sufficiently narrow such that the frequency response characteristics of the subchannels are ideal.

Without loss of generality, assume the data rates for each subcarrier are chosen from a set of N distinct data rates, denoted by $\mathscr{R}$, i.e.,

$$\mathscr{R} = \{R_0 = R_{\max}, \ldots, R_i, \ldots, R_{N-1} = R_{\min}\} \text{ (bps)} \qquad ||\mathscr{R}|| = N \tag{10.1}$$

where $R_{\min}$ and $R_{\max}$ denotes the minimum and maximum available data rates for each subcarrier, respectively. Furthermore, assume the channel frequency response is such that (in the absence of jamming) $R_{\max}$ is feasible for all subchannels.[1] Obviously, to maximize the throughput of the wireless link (or equivalently, to maximize the average data rate of OFDM symbols), the maximum achievable rate, $R_{\max}$, must be used for all subcarriers. However, because of jamming, the subchannels' capacities are not known in advanced and therefore it is not known which rates are feasible prior to transmission.

To overcome this problem, the transmitter randomly assigns the data rates to the subcarriers such that the overall throughput of the wireless link is maximized. Let the column vector $\mathbf{r}^{(n)}_{K\times 1}$ denote the transmitter's strategy for the nth OFDM symbol,

$$\mathbf{r}^{(n)}_{K\times 1}{}^{T} = [r_1 \; \ldots \; r_k \; \ldots \; r_K] \qquad \text{where } r_k \in \mathscr{R} \tag{10.2}$$

that is, for the nth OFDM symbol, the kth element of $\mathbf{r}^{(n)}$ is the data rate at which the kth subcarrier is to be coded/modulated. For each OFDM symbol, the adaptation

[1] This assumption is not particularly restrictive since any infeasible data rate can be removed from $\mathscr{R}$.

block selects a vector of K data rates where each rate is selected from the set of available data rates given in (10.1) and passes this vector to the coding/modulation block (see Fig. 10.1). The coding/modulation block transmits the nth OFDM symbol according to $\mathbf{r}^{(n)}$. From this point forward, all strategy vectors are assumed to be per OFDM symbol (i.e., for the nth symbol), and for convenience, we drop the $^{(n)}$ from the vectors.

Because of jamming, reliable recovery of the transmitted data is not guaranteed for all subchannels therefore, the resulting bit rate per OFDM symbol can be written as

$$\text{bit rate/symbol} = \sum_{k=1}^{K} \hat{r}_k \qquad \text{where } \hat{r}_k \triangleq \begin{cases} r_k, & \text{if } r_k \leq c_k \\ 0, & \text{if } r_k \geq c_k \end{cases} \tag{10.3}$$

and c_k, $k = 1, \ldots, K$ denotes the actual channel capacity for the kth subchannel, which in general, is a function of channel frequency response and the jammer power spectral density. For sufficiently narrow Δf, we can assume that the jammer's power spectral density is flat for all subchannels and therefore, we may express c_k as

$$c_k = C(p_k, j_k, N_k) \qquad \text{for} \quad k = 1, \ldots, K \tag{10.4}$$

where $C(p_k, j_k, N_k)$ denotes the channel capacity function of the wireless link and p_k, j_k and N_k (all in W/Hz) denote the power spectral densities of the transmitter, jammer and channel noise for the kth subchannel (all measured at the receiver front end), respectively.

Let $\hat{x}_i$ denote the number of subcarriers coded/modulated with $R_i \in \mathscr{R}$, $i = 0, \ldots, N-1$, obviously we have

$$\sum_{i=0}^{N-1} \hat{x}_i = K \qquad \text{and} \qquad 0 \leq \hat{x}_i \leq K \tag{10.5}$$

now let $x_i \triangleq \frac{1}{K}\hat{x}_i$, i.e., x_i denotes the fraction of subcarriers transmitted at rate R_i. Form (10.5) it follows,

$$\sum_{i=0}^{N-1} x_i = 1 \qquad \text{and} \qquad 0 \leq x_i \leq 1 \tag{10.6}$$

As a result, for sufficiently large K, the following vector can be well approximated by the following probability vector

$$\mathbf{x}_{1\times N}^{T} = [x_0 \ \ldots \ x_i \ \ldots \ x_{N-1}] \qquad \text{for} \quad K \gg N \tag{10.7}$$

If we assume that the subchannels have nearly same channel characteristics or when the effects of nonideal wireless channel (including different channel gains for subcarriers etc.) have been compensated by appropriate transmission power allocation at the receiver then, the probability vector in (10.7) can be used as an alternative way of representing the transmitter's strategy. More specifically, the following two vectors may be used interchangeably, to study the optimal transmission strategy and average throughput of the adaptive OFDM system under jamming,[2]

$$\mathbf{r}^T_{1\times K} = [r_1 \ \dots \ r_k \ \dots r_K] \ r_k \in \mathscr{R} \ \overset{\text{or}}{\longleftrightarrow} \ \mathbf{x}^T_{1\times N}$$
$$= [x_0 \ \dots \ x_i \ \dots \ x_{N-1}] \text{ s.t.} \begin{cases} 0 \leq x_i \leq 1 \\ \sum\limits_{i=0}^{N-1} x_i = 1 \end{cases} \tag{10.8}$$

That is, by assuming that the wireless channel impairments have been compensated by the proper transmission power allocation, all subcarriers experience nearly the same channel characteristics across the entire frequency band and as a result, knowing $\mathbf{x}_{N\times 1}$ which gives the fractions of subcarriers coded/modulated at available data rates is sufficient to study the performance of the wireless OFDM system under jamming.

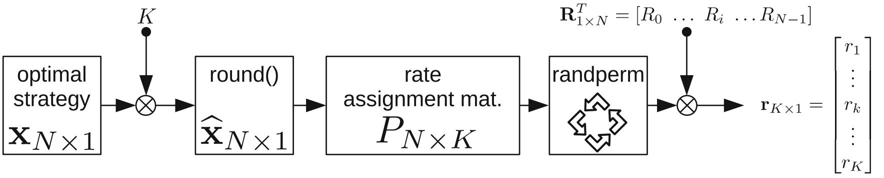
Even though the optimal transmission strategy can be computed in terms of $\mathbf{x}$, vector $\mathbf{r}$, which contains the actual transmission rates, must be reconstructed from $\mathbf{x}$ in order to code/modulate the input data. Below, we will discuss one possible approach to construct the transmission rate vector from the probability vector. First, the transmitter constructs vector $\widehat{\mathbf{x}}_{N\times 1}$ which contains the number of subcarriers coded/modulated with the available data rates. This can be done by simply multiplying the probability vector $\mathbf{x}$ by K and rounding off the results to the closest integer such that the sum remains K, i.e.,

$$\widehat{\mathbf{x}}^T_{1\times N} = \text{round } K\mathbf{x}^T_{1\times N} \tag{10.9}$$

where round() denotes the rounding to the nearest integer operation (such that the sum remains K). From vector $\widehat{\mathbf{x}}$ the *data rate assignment matrix*, denoted by $P_{N\times K}$ is constructed. To fill out the elements of P, we simply need to assign ones and zeros to the elements of P such that,

$$\sum_{k=1}^{K} P_{ik} = \hat{x}_i \qquad \text{for} \quad i = 0, \dots, N-1$$
$$\sum_{i=1}^{N} P_{ik} = 1 \qquad \text{for} \quad k = 1, \dots, K \tag{10.10}$$

[2] Also see the discussion in Sect. 10.3 for more details.



K
$\mathbf{R}^T_{1\times N} = [R_0 \ \dots \ R_i \ \dots R_{N-1}]$
optimal strategy
$\mathbf{x}_{N\times 1}$
round()
$\widehat{\mathbf{x}}_{N\times 1}$
rate assignment mat.
$P_{N\times K}$
randperm
$\mathbf{r}_{K\times 1} = \begin{bmatrix} r_1 \\ \vdots \\ r_k \\ \vdots \\ r_K \end{bmatrix}$

Fig. 10.2 The transmitter's adaptation block

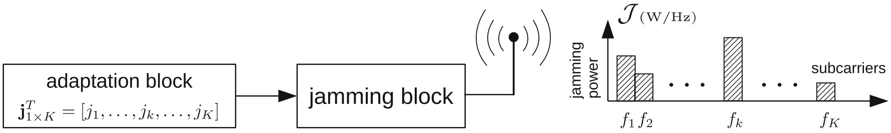


Fig. 10.3 The jammer model

That is, the number of ones in each row of P is equal to the corresponding component of $\widehat{\mathbf{x}}$ and every column contains exactly one 1. For example, one possible data rate assignment matrix is given in (10.11).

$$P_{N\times K} = \begin{bmatrix} 111\ 00\ 0000 \dots \\ 000\ 11\ 0000 \dots \\ 000\ 00\ 1111 \dots \\ \dots \end{bmatrix} \begin{matrix} \leftarrow \#\text{ of } 1'\text{s} = \hat{x}_0 \\ \leftarrow \#\text{ of } 1'\text{s} = \hat{x}_1 \\ \leftarrow \#\text{ of } 1'\text{s} = \hat{x}_2 \\ \end{matrix} \tag{10.11}$$

A 1 at row i and column k of P, indicates that the kth subcarrier is to be coded/modulated with the ith data rate, Finally, the rate vector for the nth OFDM symbol, $\mathbf{r}^{(n)}_{K\times 1}$, can be written as

$$\mathbf{r}^{(n)}_{1\times K}{}^{T} = \mathbf{R}^T_{1\times N} \text{ randperm}^{(n)} \ P_{N\times K} \tag{10.12}$$

where $\mathbf{R}$ is the vector of available data rates (from (10.1)) and randperm$^{(n)}P_{N\times K}$ denotes randomly permuting columns of $P_{N\times K}$ for n times. As we discuss in Sect. 10.3, for each OFDM symbol, it is necessary to randomly permute columns of P. Figure 10.2 shows the transmitter's adaptation block.

10.2.2 Jammer Adaptivity Model

The jammer has an adaptation block which allows him to jam individual subchannels with (possibly) different jamming powers (see Fig. 10.3). However, for practical reasons, the jammer's maximum jamming power per subchannel is limited to $J_{\max}$ (W/Hz). Furthermore, we assume the jammer's energy budget per OFDM symbols is limited to $E_{\max}$ (Joules). Finally, we assume $E_{\max}$ is such that the jammer cannot use $J_{\max}$ for all subchannels otherwise, the energy constraint would be redundant.

We denote the jammer's jamming power set by $\mathscr{J}$. Without loss of generality assume, $\mathscr{J}$ is given as follows,

$$\mathscr{J} = \{J_0 = 0, \ldots, J_i, \ldots, J_{M-1} = J_{\max}\} \text{ (W/Hz)} \qquad \text{where} \quad ||\mathscr{J}|| = M \tag{10.13}$$

Because of the jammer's energy constrained, $E_{\max}$, the jammer's average power, denoted by $\mathrm{E}[J]$, is constrained to

$$\mathrm{E}[J] \leq \frac{E_{\max}}{T_s} \tag{10.14}$$

where T_s is the OFDM symbol duration. Let $\mathbf{j}_{K\times 1}$ denote the jammer's strategy per OFDM symbol (equivalently, jamming vector),

$$\mathbf{j}^T_{1\times K} = [j_1 \ \ldots \ j_k \ \ldots \ j_K] \text{ (W/Hz)} \qquad \text{where} \quad j_k \in \mathscr{J} \tag{10.15}$$

that is, the kth subchannel is jammed with $j_k \in \mathscr{J}$ (W/Hz). From (10.14) we have,

$$T_s \sum_{k=1}^{K} j_k \Delta f = T_s \Delta f \sum_{k=1}^{K} j_k \leq E_{\max} \tag{10.16}$$

$$\Rightarrow \qquad \sum_{k=1}^{K} j_k \leq \frac{E_{\max}}{T_s \Delta f} \tag{10.17}$$

Now let $\hat{y}_i$ denote the number of subchannels jammed with $J_i \in \mathscr{J}$, obviously, we have

$$\sum_{k=1}^{K} j_k = \sum_{i=0}^{M-1} \hat{y}_i J_i \tag{10.18}$$

It is also clear that,

$$\sum_{i=0}^{M-1} \hat{y}_i = K \qquad \text{and} \qquad 0 \leq \hat{y}_i \leq K \tag{10.19}$$

Define $y_i \triangleq \frac{1}{K}\hat{y}_i$, from (10.19), we have

$$\sum_{i=0}^{M-1} y_i = 1 \quad \text{and} \qquad 0 \le y_i \le 1, \qquad \text{for } i = 0, \ldots, M-1 \tag{10.20}$$

Therefore, for sufficiently large K, the following vector can be well approximated by a probability vector

$$\mathbf{y}_{1\times M}^{T} = [y_0 \ \ldots \ y_i \ \ldots \ y_{M-1}] \qquad \text{for} \quad K \gg M \tag{10.21}$$

If we write (10.17) in terms of y_i, $i = 1, \ldots, M$, we obtain the following constraint on $\mathbf{y}$.

$$\sum_{k=1}^{K} j_k = K \sum_{i=1}^{M} y_i J_i \le \frac{E_{\max}}{T_s \Delta f} \tag{10.22}$$

Therefore, (10.21) can be used as an alternative way of representing the jammer's strategy. More specifically, the following two vectors can be used interchangeably to study the optimal jamming strategy and average performance degradation of the wireless link[3]

$$\begin{aligned} \mathbf{j}_{1\times K}^{T} = [j_1 \ \ldots \ j_k \ \ldots j_K]\,, \ j_k \in \mathscr{J} \ \overset{\text{or}}{\longleftrightarrow} \ & \mathbf{y}_{1\times M}^{T} = [y_0 \ \ldots \ y_i \ \ldots \ y_{M-1}] \\ \text{s.t.} \quad & \begin{cases} 0 \le y_i \le 1 \\ \displaystyle\sum_{i=0}^{M-1} y_i = 1 \\ \displaystyle\sum_{i=0}^{M-1} y_i J_i = \mathbf{y}^T \mathbf{J} \le \frac{E_{\max}}{K T_s \Delta f} \end{cases} \end{aligned} \tag{10.23}$$

Even though the optimal jamming strategy can be computed in terms of $\mathbf{y}$, as it is shown in Fig. 10.3, vector $\mathbf{j}^{(n)}$, which contains the actual jamming powers for the nth OFDM symbol, must be passed to the jamming block in order to allocate the available jamming power accordingly. The jamming vector can be constructed from the jamming probability vector in the exact same manner that the rate vector was constructed. Here we only provide the results and refer the reader to Sect. 10.2.1 for the details.

$$\mathbf{j}_{1\times K}^{(n)\ T} = \mathbf{J}_{1\times M}^{T} \ \text{randperm}^{(n)} \ P_{M\times K} \tag{10.24}$$

[3] See the discussion in Sect. 10.3 for more details.

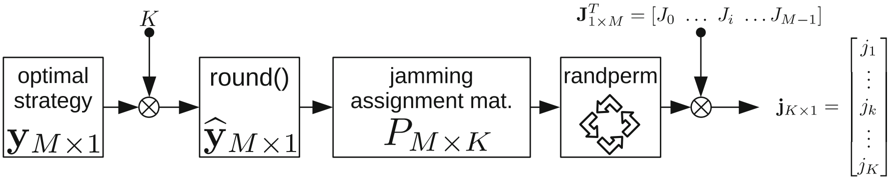

K
optimal strategy $\mathbf{y}_{M\times 1}$
round() $\hat{\mathbf{y}}_{M\times 1}$
jamming assignment mat. $P_{M\times K}$
randperm
$\mathbf{J}^T_{1\times M} = [J_0 \ \dots \ J_i \ \dots J_{M-1}]$
$\mathbf{j}_{K\times 1} = \begin{bmatrix} j_1 \\ \vdots \\ j_k \\ \vdots \\ j_K \end{bmatrix}$

Fig. 10.4 The jammer's adaptation block

where $P_{M\times K}$ denotes the jamming assignment matrix, and is constructed in the exact same way as (10.11) was constructed, and randperm$^{(n)}$ denotes randomly permuting columns of P for n times. Figure 10.4 shows the jammer's adaptation block in detail.

The random permutation block is necessary in the jammer's adaptation block to randomize the jamming vector for every OFDM symbol before passing it to the jamming block. Otherwise, the subchannels will be jammed with a fixed power and the jamming problem simplifies to a simple water filling problem [8].

10.3 Proactively Optimizing the Average Throughput of Adaptive OFDM in the Presence of Adaptive Jammers

We consider the problem of maximizing the average throughput of the wireless link under jamming by randomly adapting the data rates of the subcarriers.[4] Obviously, in absence of the jammer, the optimal strategy to maximize the average throughput is to use the maximum data rate for all subcarriers. However, in the presence of the jammer, the capacities of the subcarriers are not known in advance.

By assigning different data rates to the subcarriers, the transmitter can increase the possibility that some of the subcarriers overcome the jamming (see Fig. 10.5). Furthermore, data rates assignments must be done randomly, that is, for every OFDM symbol the data rate assignment pattern must be randomized. This randomization is necessary since static data rate assignment pattern would make higher data rates more vulnerable to jamming as these data rates are easier to jam.

Consider a typical wireless OFDM system such as IEEE 802.11, without loss of generality, assume the set of available data rates in the OFDM system, $\mathscr{R}$ is sorted in a decreasing order, i.e.,

$$\mathscr{R} = \{R_0 = R_{\max}, \dots, R_i, \dots, R_{N-1} = R_{\min}\} \text{ bps} \qquad (10.25)$$

[4] This is equivalent to maximizing the average number of data bits per OFDM symbol or the average data rate per OFDM symbol.

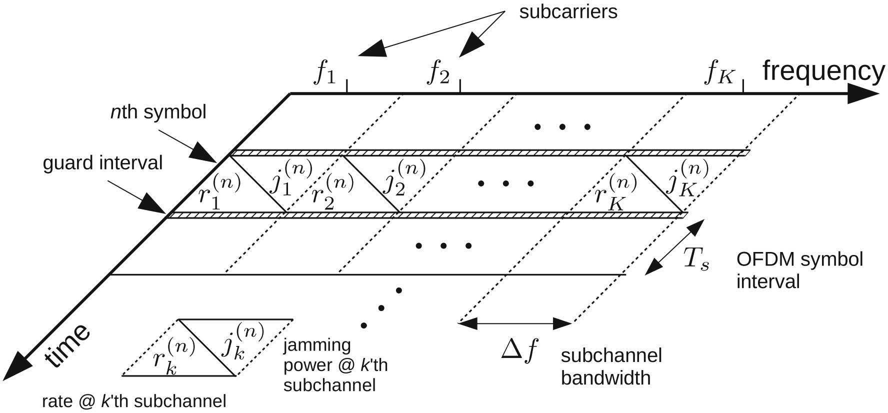


Fig. 10.5 Adaptive OFDM under jamming

where R_i, $i = 0, \ldots, N$ are the available data rates of the OFDM system for the subcarriers. It can be shown that to jam a rate adaptive wireless system with N rates, it sufficient to use no more than $N + 1$ jamming powers [6]. Therefore, WLOG, we assume the jammer is using the following jamming set

$$\mathscr{J} = \{J_0 = 0, \ldots, J_j, \ldots, J_N = J_{\max}\} \text{ W/Hz} \tag{10.26}$$

When the wireless channel is nearly flat or, when the jamming is the dominant cause of the noise at the receiver front end (which is the typical case for most jamming scenarios), it can be assumed that each rate in $\mathscr{R}$ can tolerate up to a certain level of jamming power which is the same for all subchannels and it is completely lost otherwise (since every subchannel experiences nearly the same channel or the jamming is the dominant factor of the noise). As we will see shortly, this assumption greatly simplifies the analytical results and allows us to express the results in closed form expressions.

Assume $R_i \in \mathscr{R}$, $0 \le i \le N-1$, can be recovered for any jamming power less than J_{i+1}, $0 \le i \le N-1$. That is, R_0 can only tolerate J_0 and no rate can tolerate J_N. Furthermore, let the transmitter's and the jammer's strategies be,[5]

$$\hat{\mathbf{x}}^T = \left[\hat{x}_0 \ \ldots \ \hat{x}_i \ \ldots \ \hat{x}_{N-1}\right]_{1\times N} \qquad \hat{x}_i : \text{\# of subchannels to be sent with } R_i \tag{10.27}$$

and

[5] Throughout the rest of this chapter, whenever clear from the context, we refer to the vectors **r**, **x** and $\hat{\mathbf{x}}$ as the transmitter's strategies interchangeably. In other cases, we explicitly mention which of the vector are being referred to.

$$\widehat{\mathbf{y}}^T = \begin{bmatrix} \hat{y}_0 & \ldots & \hat{y}_i & \ldots & \hat{y}_N \end{bmatrix}_{1\times(N+1)} \qquad \hat{y}_i : \text{\# of subchannels jammed with } J_i \tag{10.28}$$

where $\widehat{\mathbf{x}}$ and $\widehat{\mathbf{y}}$ are defined and constructed as discussed in Sects. 10.2.1 and 10.2.2, respectively. Since the transmitter and the jammer randomize their respective strategies independently, the partial average throughput from rate R_0, and denoted by T_0, becomes

$$T_0 = \hat{x}_0 \left(\frac{\hat{y}_0}{K} \right) R_0 \tag{10.29}$$

that is because $\hat{x}_0$ is the number of subcarriers coded/modulated with R_0 and $\hat{y}_0/K$ is the probability that a subchannel is jammed with J_0. Similarly, the partial average throughput from data rate R_i is

$$T_i = \hat{x}_i \left(\frac{1}{K} \sum_{j=0}^{i} \hat{y}_j \right) R_i \tag{10.30}$$

Therefore, the average throughput per OFDM symbols becomes,

$$\begin{aligned} T &= \sum_{i=0}^{N-1} T_i = \sum_{i=0}^{N-1} \hat{x}_i \left(\frac{1}{K} \sum_{j=0}^{i} \hat{y}_j \right) R_i \\ &= K \sum_{i=0}^{N-1} \sum_{j=0}^{i} \frac{\hat{x}_i}{K} \frac{\hat{y}_j}{K} R_i \\ &= K \sum_{i=0}^{N-1} \sum_{j=0}^{i} x_i y_j R_i \\ &= \mathbf{x}^T K R_{N\times(N+1)} \mathbf{y} \end{aligned} \tag{10.31}$$

where, $\mathbf{x}$ and $\mathbf{y}$ are defined in (10.7) and (10.21), respectively and the matrix $R_{N\times(N+1)}$ is a lower triangular matrix where the nonzero elements of the rows of R are equal to the data rates, i.e.,

$$R_{N\times(N+1)} = \begin{bmatrix} R_0 & 0 & & \ldots & & 0 \\ \vdots & \ddots & \ddots & & & 0 \\ R_i & & R_i & 0 & \ldots & \vdots \\ \vdots & & & \ddots & \ddots & \\ R_{N-1} & & \ldots & & R_{N-1} & 0 \end{bmatrix} \tag{10.32}$$

therefore, the optimal transmission/jamming and the average throughput of the wireless link at the equilibrium is the solution of the following maxmin problem,

$$T(\mathbf{x}^*, \mathbf{y}^*) = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T K R_{N\times(N+1)} \mathbf{y} \quad \text{s.t.} \quad \begin{cases} \mathbf{x}^T \mathbf{1} = 1 \\ \mathbf{y}^T \mathbf{1} = 1 \\ \mathbf{y}^T \mathbf{J} \leq \dfrac{E_{\max}}{K T_s \Delta f} \end{cases} \tag{10.33}$$

where, $\mathbf{x}^*$ and $\mathbf{y}^*$ denote the optimal transmission and jamming strategies, respectively. The maxmin problem in (10.33) can be solve analytically and numerically (Sect. 10.4).

10.4 Generalized Interactions: Constrained Bimatrix Games

In zero-sum game framework, it is usually assumed that the players have *perfect knowledge* of the game and the actions that are available to the other players, and they use this knowledge to compute their respective optimal strategies. In such a case, the zero-sum framework fully captures the conflicting goals of the players. Moreover, the equilibrium solution of the zero-sum game guarantees a minimum payoff regardless of the other player's strategy [9]. For a more comprehensive study of constrained zero-sum games in wireless jamming see [6].

However, in some jamming scenarios, having perfect knowledge of the system parameters (or available actions) may not be a feasible option or too costly for a player. In addition, players may have objectives that are not exactly the opposite of each other, for example, the transmitter may wish to minimize the average error probability while the jammer wishes to minimize the average throughput of the system (as opposed to maximizing the average error probability).

In such scenarios, a more appropriate framework to model the communication system under jamming would be a *bimatrix* game instead of a zero-sum game.[6] In bimatrix games it is no longer required that the sum of the players' payoffs to be zero (or a constant value) [9]. As a result, players can have different objectives and the respective payoffs can be defined based on the players' goals and their knowledge of the game (which in general may be imperfect). Such a formulation, encompasses a variety of situations from full competition to full cooperation.

Additionally, in standard zero-sum and bimatrix games there are no additional restrictions on players' mixed-strategies, i.e., players may choose any probability distribution over their respective action sets (pure-strategies). However, there exist

[6] It can be shown that zero-sum games are special cases of the more general bimatrix games.

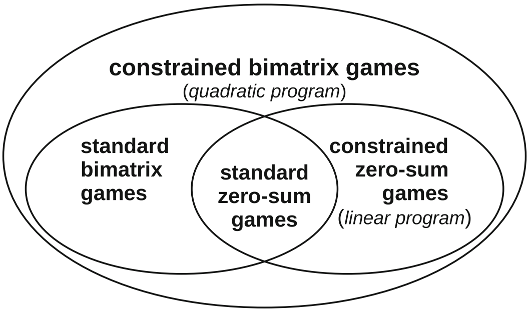


Fig. 10.6 Classification of standard and constrained games

scenarios where, due to practical reasons, not all mixed-strategies are permitted and/or feasible.

Such scenarios demand for a more general framework to study them. In this chapter, we study a *linearly constrained bimatrix* game to overcome these limitations. In constrained games, the players' mixed-strategies not only have to be a probability distributions but they must satisfy some additional linear constraints too[7] (Fig. 10.6 shows the classification of standard and constrained games). Detailed study of necessary and sufficient conditions under which the existence of the Nash equilibrium is guaranteed as well as a systematic approach to find the NE is beyond the scope of this chapter, but can be found in [7].

10.5 Performance Analysis: The Case of IEEE 802.11 OFDM

In IEEE 802.11, the 20 MHz channel bandwidth is subdivided into 52 subchannel with a separation of $\Delta f = 20/64 = 0.3125$ MHz. Of the total 52 subcarriers, 48 carry data and 4 are pilot subcarriers, however, for convenience, in our analysis we assume the number of subcarriers is $K = 64$ and all the subcarriers carry data. The OFDM symbol interval is $T_s = 4$ µs (which includes a 0.8 µs cyclic prefix), which results in the symbol rate $R_s = 0.25$ MSymbols/s. Table 10.1 shows some of the IEEE 802.11 physical layer parameters including the modulation schemes and code rates used for the subcarriers. The last column in Table 10.1 shows the resulting data rates for the subcarriers. Hence, the set of available data rates for the subcarriers is

$$\mathscr{R} = \{R_0 = 1.125,\ R_1 = 1.0,\ 0.75,\ 0.5,\ 0.375,\ 0.25,\ 0.1875,\ R_7 = 0.125\}_{\text{Mbps}} \tag{10.34}$$

[7] In this paper, we limit out focus to linear constraints on players' strategy sets and for convenience, we use the terms *constrained games* and *linearly constrained games* interchangeably.

Table 10.1 IEEE 802.11 modulation schemes and code rates

Modulation	Code rate	Bits/subcarrier	Data rate/subcarrier
BPSK	1/2	1/2	125 Kbps
	3/4	3/4	187.5 Kbps
QPSK	1/2	1	250 Kbps
	3/4	3/2	375 Kbps
16-QAM	1/2	2	500 Kbps
	3/4	3	750 Kbps
64-QAM	2/3	4	1 Mbps
	3/4	9/2	1.125 Mbps

Suppose that we have a base station that is communicating with a mobile user and the total transmission power for the 64 subcarriers is 200 mW. Then transmission power per subcarrier becomes,

$$P_T = \frac{0.2}{64} = 0.0031 \text{ W} = -25 \text{ dBW} \tag{10.35}$$

Furthermore, assume that the combined transmitter/receiver antenna gains and the losses from other sources result in a $L_T = 90$ dB attenuation in the received signal power. Then, the power of the received signal per subcarrier is,

$$(P_R)_{\text{dB}} = (P_T)_{\text{dB}} - (L_T)_{\text{dB}} = -25 - 90 = -115 \text{ dBW} \tag{10.36}$$

The power spectral density of additive noise at the receiver front end is $N_0 = 4.1 \times 10^{-21}$ W/Hz. Therefore the signal to noise ratio at the receiver front end becomes,

$$\text{SNR}_{\text{dB}} = (P_R)_{\text{dB}} - (\Delta f N_0)_{\text{dB}} \cong 34 \tag{10.37}$$

and from (10.37) the capacity of the subcarriers is

$$C_{\text{AWGN}} = \Delta f \log(1 + \text{SNR}) = 3.52 \text{ Mbps} \tag{10.38}$$

Since the capacity of the link is greater than $\max_i \mathbf{R}$, all the available data rates are feasible (when the jammer is not active). Now, suppose that the jammer's combined antenna gain and losses from other sources results in a $L_J = 60$ dB attenuation in the jamming signal measured at the receivers front end. To make the channel capacity drop below the data rate (i.e., make the data rates infeasible), the jammer needs to increase the channel noise at the receiver front end by

$$P_{JR,i} = \Delta f J_i = \frac{P_R}{2^{R_i/\Delta f} - 1} - \Delta f N_0 \quad \text{(W)} \tag{10.39}$$

where R_i's are given in (10.34). Hence, the jammer's transmission power becomes

$$(P_{JT,i})_{\mathrm{dB}} = (P_{JR,i})_{\mathrm{dB}} + (L_J)_{\mathrm{dB}} \tag{10.40}$$

If we substitute the numerical values, the jammer's strategy set becomes

$$\begin{aligned}\mathscr{J} &= \{0, 0.0003, 0.0004, 0.0007, 0.0015, 0.0024, 0.0042, 0.0061, 0.0098\} \\ &\quad \mathrm{W/subchannel} \\ &= \{-\infty, -35.2, -34.2, -31.4, -28.1, -26.2, -23.8, -22.2, -20.1\} \\ &\quad \mathrm{dBW/subchannel}\end{aligned} \tag{10.41}$$

The optimal transmission and jamming strategies and the expected value of the game at the Nash equilibrium can be derived analytically (and numerically) in the same manner as was presented in Sect. 10.4. For Instance, it can be shown that the minimum average jamming power needed to force the lowest rate for the jammer that is using optimal strategy is

$$J_{\mathrm{TH}} = K\Delta f R_7 \sum_{i=1}^{7} \left(R_i^{-1} - R_{i-1}^{-1}\right) J_j = 0.2133 \text{ W} = -6.7 \text{ dBW} \tag{10.42}$$

whereas the average jamming power that the Barrage noise jammer require to achieve the same average throughput is

$$J_{\mathrm{barrage}} = \frac{K P_R}{2^{R_{\min}/K\Delta f} - 1} - K\Delta f N_0 = 0.39 \text{ W} = -4.1 \text{ dBW} \tag{10.43}$$

This shows a gain of 2.6 dB for the strategic jammer.

10.6 Concluding Remarks

In this chapter, we studied the performance of an adaptive OFDM wireless communication system under power limited jamming. We showed that with modest assumptions, this problem can be formulated into the constrained zero-sum or constrained bimatrix game. We in particular applied this framework to the IEEE 802.11 with OFDM physical layer.

References

1. C. Shahriar, M. La Pan, M. Lichtman, T.C. Clancy, R. McGwier, R. Tandon, S. Sodagari, J.H. Reed, PHY-layer resiliency in OFDM communications: a tutorial. IEEE Commun. Surv. Tutorials **17**(1), 292–314 (2015)
2. T.C. Clancy, N. Goergen, Security in cognitive radio networks: threats and mitigation, in *3rd International Conference on Cognitive Radio Oriented Wireless Networks and Communications, 2008. CrownCom 2008* (IEEE, Piscataway, 2008), pp. 1–8
3. M. Pan, T.C. Clancy, R.W. McGwier, Jamming attacks against OFDM timing synchronization and signal acquisition, in *Military Communications Conference, 2012-MILCOM 2012* (IEEE, Piscataway, 2012), pp. 1–7
4. T.D. Vo-Huu, T.D. Vo-Huu, G. Noubir, Interleaving jamming in wi-fi networks, in *Proceedings of the 9th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, WiSec '16 (2016), pp. 31–42
5. E. Altman, K. Avrachenkov, A. Garnaev, A jamming game in wireless networks with transmission cost, in *1st EuroFGI International Conference on Network Control and Optimization, NET-COOP* (2007), pp. 1–12
6. K. Firouzbakht, G. Noubir, M. Salehi, On the performance of adaptive packetized wireless communication links under jamming. IEEE Trans. Wirel. Commun. **13**(7), 3481–3495 (2013)
7. K. Firouzbakht, G. Noubir, M. Salehi, Linearly constrained bimatrix games in wireless communications. IEEE Trans. Commun. **64**(1), 429–440 (2016). https://ieeexplore.ieee.org/abstract/document/7339449
8. M. Salehi, J. Proakis, *Digital Communications* (McGraw-Hill Higher Education, McGraw-Hill Education, Boston, 2007)
9. G. Owen, *Game Theory* (Academic, San Diego, 1995)