Attack and Defense Strategies in Cyber-Physical Systems with Varying Levels of System and Opponent Knowledge

Submitted in partial fulfillment of the requirements for

the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

Bruce E. DeBruhl II

M.S. Electrical and Computer Engineering, Carnegie Mellon University
B.S. Electrical Engineering, Kettering University

Carnegie Mellon University
Pittsburgh, PA

May, 2015

Abstract

Advances in computing, communications, and sensing have enabled exciting opportunities for large scale applications of cyber-physical systems (CPS) to energy, transportation, healthcare, and defense. All of these services support critical applications, making CPS security crucial. For example, an attack against the smart-grid, or a power grid enhanced with CPS, may result in devastating regional blackouts. Fortunately, the technologies that enable CPS allows us to design attack and defense strategies leveraging robust sensing and actuation.

In this thesis, we explore the interaction of two adversarial players with a shared cyber-physical system. We investigate how a player with limited information about the CPS or their opponent chooses an attack or defense. In particular, we explore the following question: *how is an agent's strategy affected by the amount of knowledge they have about the CPS they interact with and their opponent's strategy?*

We consider various scenarios to explore this problem including: an agent that interacts with a known system and known opponent, an agent that interacts with a known system and an opponent with assumed behavior, an agent that interacts with a known system and an unknown opponent, and an agent that interacts with a known opponent and a partially known system. For each of these scenarios we provide a proof-of-concept attack or defense to demonstrate security challenges and opportunities. We also introduce other scenarios based on system and opponent knowledge levels that demonstrates exciting future research opportunities.

Acknowledgments

There have been many people that have supported and encouraged me throughout my PhD studies. To all of these people I owe a sincere debt of gratitude and extend a heart felt **Thank You!**

I am very grateful to have worked with my adviser and committee chair, Patrick Tague. I am thankful for your mentoring and guidance on how to find and approach challenging research problems. I am also very thankful for your example as an academic who balances work and family and genuinely cares about your students' well-being.

I am very thankful to my dedicated committee members who have guided me through the thesis writing process. These committee member include Anupam Datta, Bob Iannucci, and Kostas Pelechrinis. I am deeply indebted not only for your guidance and support in writing this thesis but also for your guidance in other endeavors.

I would like to thank mentors that have helped me to hone my skills through the PhD program including Abe Ishihara, Bruno Sinopoli, Ed Katz, George Nychis, Ole Mengshoel, Pei Zhang, and Thomas Sandholm. I am extremely grateful for the knowledge you have shared with me about diverse engineering topics.

My choice to pursue a PhD at CMU was the end result of many peoples guidance and encouragement. I am very thankful to JC Irvine for introducing me to computing and electronics. I am also thankful to Nozar Tabrizi and Huseyin Hiziroglu for their guidance on engineering fundamentals and problem solving at Kettering University. I am particularly indebted to Nozar for encouraging me to apply to a PhD program.

I am thankful to have shared this journey with an amazing group of labmates: Yu Seung,

Arjun, Brian, Sean, Yuan, Le, Eric, Ervin, and Jun. I have enjoyed working with each of you and look forward to continued friendship for years to come. I am also thankful to amazing student collaborators including Michelle, Zack, Christian, and Sean. It was a pleasure to work and learn from each of you on interesting topics.

I would like to thank the National Defense Science and Engineering Graduate Fellowship for their financial support to allow me to freely pursue my PhD.

I have been supported by amazing friends throughout this program and am very thankful for their support. There have been many friends that have been willing to hear about what I am working on and encourage me when things were not working, which is a lot of the time in research.

I owe a great debt to my loving parents, Bruce and Lorie, for all there support. I am very thankful for their support and encouragement. I would like to thank the rest of my family including my Sister, Brother-in-law, and my niece and nephews for their support and encouragement.

I am deeply blessed by my wonderful wife, Heather, for encouraging, supporting, and challenging me. Her support has kept me going when I felt defeated, I could not have done this without her.

Last but not least, I would like to thank God for His grace and sustenance throughout the whole PhD process. ”Now to our God and Father be the glory forever and ever. Amen.“ - Philippians 4:20.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Advances in cyber-physical systems (CPS) allow using sensing, actuation, computation, and networking to accomplish complex tasks effectively and efficiently. CPS enables new techniques for managing scarce resources, providing enhanced services, and empowering large-scale data science. For example, applying CPS to the power-grid allows efficient use of on-demand electric plants, on-line detection of dangerous failures, and real-time consumer feedback. Many other domains can be enhanced by CPS including transportation, e-health, defense, and smart-cities.

Since many applications of CPS support critical infrastructure it is essential to explore cyber-physical security (CPSec) [1]. Researching CPSec involves both the design of tools to make CPS robust to attacks and investigating potential attack vectors. For example, it is important to understand what security breaches in the smart-grid can cause regional blackouts and how to prevent them. Conversely, it is important to understand the breadth of attacks and their motivations. For example, an attack can be designed against vehicle formations to cause accidents that result in loss-of-life or high-dollar property damage. An attack against vehicle formations can also be more benign, for instance an attack may cause reduced efficiency [2]. One high-profile in-the-wild CPS attack was the STUXNET virus [3] which caused nuclear centrifuges to spin at unsafe speeds while spoofing normal sensor readings to the operator. This attack, whose origin has been difficult to trace, caused damage to Iranian nuclear reactors.

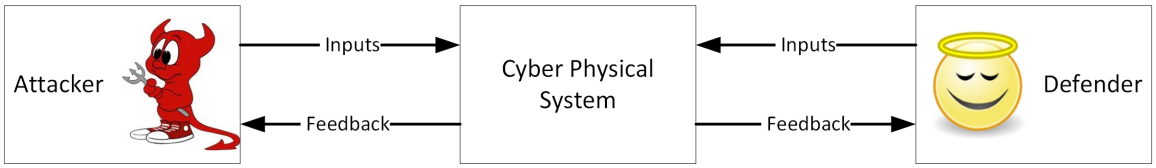


Figure 1.1: In this figure, we show our model consisting of two competing players that interact with a common cyber-physical system. In this thesis, we consider how to design a player's strategy based on limited information about the system and their opponent's strategy.

Fortunately, We can use the technologies that enable CPS to design interesting and unique attacks and defenses that leverage network statistics and physical behavior. For example, we can design an attacker that mounts network attacks (denial-of-service, disassociation), control attacks (destabilizing the system, malicious control algorithms), or a combination thereof. More interestingly, we can design an attacker that uses cross-domain attacks. For instance, we can consider an attack that uses denial-of-service to destabilize the physical system's controller. Similarly, CPS enables us to design defenses that use combinations of physical measurements and network statistics to infer when an attack is being mounted.

In this thesis, we consider the two-player CPS security model shown in Figure 1.1. This model contains two competing players, or agents, that interact with a common CPS. We define competing to mean that the two players have one, or more, opposing objectives. Throughout this thesis, we refer to the malicious and benign player as the attacker and defender respectively. Both players in our model control an input signal and receive feedback from the system through sensing. The system is cyber-physical so it contains both physical and digital states that are updated based on the current system state and both player's input signal.

We are interested in exploring attack and defense strategies based on what a player knows about their opponent and the system. In particular, how does a player in our 2-player CPS model select a security strategy given limited knowledge of their opponent's strategy and the system. We define security strategy to mean the set of actions or algorithms that define a player's attack or defense vector. A key question in this context is thus: how does the quantity and quality of info available to each opponent affect the notion of optimality in different attack and defense

scenarios?

In realistic CPSec scenarios, a player's knowledge of the system may range from full knowledge to no knowledge. For example, a defender or an inside attacker, may have a full model of the system they are interacting with. Conversely, an external attacker may have no knowledge of the system. We define 4 levels of system knowledge that a player may have.

- *Full knowledge*: The player has knowledge of the system update equations, dynamics, parameters, etc. This allows the player to deterministically model the behavior of the system and how they impact it. This is likely the case for inside attackers or many defenders.
- *Partial knowledge with Slow-Changing Parameters.*: The player has knowledge of the system's update equations or how they interact with the system but does not know the specific system parameters. The system parameters are assumed to be constant or slow to change. This is potentially the case for a new defender or an external attacker. The assumption of slow-changing parameters is often reasonable and allows for system identification techniques to be used to design effective attack and defense strategies.
- *Partial Knowledge with Fast-Changing Parameters*: The player has knowledge of the system's update equation or how they interact with the system but does not know the specific system parameters. The system parameters may change rapidly making modeling and system identification more challenging, potentially intractable. The rapid changing parameters in this scenario often requires the use of stochastic modeling techniques.
- *No Knowledge*: The player has no previous knowledge of the system. Learning and modeling must be completed with no a priori information to determine a security strategy. This is likely the case for an outside attacker.

Any of these system knowledge levels can also be considered in a stochastic system.

Similarly, in realistic CPSec scenarios a player's knowledge of their opponent's strategy can vary widely. Since a player may or may not know their opponent's strategy we define 3 levels of knowledge that a player may have.

		System Knowledge			
		Known	Slow Parameter	Fast Parameter	Unknown
Opponent	Known	Chapter 3			
	Assumed	Chapter 5	Chapter 6		
	Unknown	Chapter 4			

Table 1.1: In this table, we illustrate the broad problem of adversarial interaction with limited knowledge in CPS. We color the problems explored in this thesis in yellow and for future work in grey.

- *Known Strategy*: The player knows deterministicly what their opponent do. Possible for an inside attacker.
- *Assumed Strategy*: The player assumes that their opponent uses a particular strategy. This includes a player assuming their opponent is rational in a game-theoretic sense, constant, or adaptive.
- *Unknown Strategy*: The player has no knowledge of their opponent's strategy. This is often the case for a defender or external attacker.

As before, we can also consider that a player has an opponent with stochastic behavior.

In this thesis, we consider choosing security strategies given various combinations of system and opponent knowledge levels. In particular, we consider the following system and opponent knowledge levels.

- In Chapter 3, we introduce an insider attack which is designed with full-knowledge of the system and full-knowledge of the defender's strategy. We highlight the devastating nature of this attack on a legitimate system.
- In Chapter 4, we investigate a technique to defend a critical CPS with no knowledge of the attacker but full knowledge of the system. We show that we are able to mitigate the impact of attacks with minimal performance cost.
- In Chapter 5, we use game theory to explore the interaction of a rational player who has full-knowledge of the system and assumes a rational opponent. We compare the perfor-

mance of rational players with constant, random, and adaptive opponents to gain a deeper understanding of trade-offs in the strategies and the impact of incorrectly assuming an opponent's strategy.

- In Chapter 6, we design an attacker that identifies the parameters of a partially known system with slow-changing parameters against an assumed constant opponent. We show that using system identification techniques allows an adaptive attacker to be very efficient and effective.

In Table 1.1 we highlight topics we explore in yellow with corresponding thesis chapters.

There are other interesting combination in Table 1.1 that we do not explore but leave as future work. Particularly, an unknown opponent and a system with unknown slow-changing parameters provides an interesting research area. It is unclear whether an unknown system with an unknown opponent is a feasible problem since differentiating the systems actions from an opponents behavior may be impossible but it opens up an extremely interesting attack space.

In the remainder of this chapter we introduce technologies that motive our work and then summarize our major contributions.

1.1 Motivating Use Cases

Over the past decade mobile technologies have rapidly advanced due to, among other reasons, the plummeting cost of computing [4], the decreasing cost of batteries, and the decreasing cost of data storage [5]. Simultaneously, advances in sensor design, data aggregation, and data processing allow for the availability of plentiful rich data about the physical world that we live in and the digital world that we depend on. Combining plentiful sensing with cheap computing opens up new possibilities for cyber-physical systems that leverage intelligent observation with agile control and actuation in applications including smart-grid, cooperative vehicular networks, smart-homes, software-defined networking, and electronic healthcare.

In general, CPS are constrained to operating conditions and limits. For example, a system that interacts with the physical world has to act within the bounds of physics and its actuators performance. A time-critical system requires that control inputs for actuators are calculated quickly, which may necessitate a good enough control instead of an optimal control. Similarly, wireless systems have fundamental limits imposed by shared bandwidth in a local area.

Since CPS is often used in critical applications like transportation, energy, healthcare, or communications their security is essential. In this section, we present two motivating use cases of CPS that we explore throughout thesis: cooperative adaptive cruise control and software-defined radio.

1.1.1 Platooning

Traffic is a growing source of frustration in most urban areas and a major source of deaths due to driver errors and inclement road conditions. The percentage of vehicle related deaths is particularly startling for people under the age of 35; according to the CDC for people in the U.S. aged 5-34, traffic accidents are a leading cause of deaths [6]. Because of this, an ever increasing body of work [7, 8, 9] has explored the use of autonomous and semi-autonomous driving, allowing for a car to pilot itself while the passenger inside can focus on other tasks.

One increasingly popular driver-assistance feature is adaptive cruise control (ACC) which uses radar to keeps a constant-headway following distance to the preceding car. The radar's intervention allows for the car to safely maintain a constant headway and reduce accidents caused by insufficient following distance. Insufficient following distance is a common occurrence in manual vehicle operation with many motorists using a headway under 1 second, and almost all motorists using a headway under the recommended 2 seconds [10].

The performance of ACC is limited by the vehicle's physical behavior with brake lag as a major contributing factor. Brake lag is the time it takes for a car to start decelerating after an electronic brake signal has been received. Brake lag is particularly bad with air brakes, which

requires time to pressurize the system before deceleration begins. With the advent of vehicle to vehicle communications we can design ACC equipped cars that collaborate to further reduce their headway speed. This is enabled because the delay for V2V communications is less then the delay from brake lag allowing lines of cars to safely decelerate while using small headway times. This type of formation driving is called cooperative ACC (CACC) or platooning. The benefits of platooning include increased density of cars on a highway and increased fuel efficiency of platooned vehicles [11].

To date, the work on platooning has largely focused on how to design a controller that is string stable. String stability is the property that the error in following distance does not grow along a platoon of vehicles [9]. While preliminary work has considered string stability in ideal systems (e.g. no networked communications [9] or perfect networks [12]), recent work has explored string stability in realistic networks including networks with delays [13], packet-based networks [14], and stochastic networks [15].

1.1.2 Software-Defined Radio for Jamming and Anti-Jamming

Software defined radio and networking (SDR/N) enable radios and networks to adapt their operations according to demand, performance, and other objectives. In this thesis, we consider in how SDR can enable agile jamming, or injection of intentionally interfering signals into the wireless medium as illustrated in Figure 1.2, and anti-jamming techniques.

While jamming has been a topic of research for several decades [16], partially due to the devastating potential and difficulty of defense, the SDR revolution has sparked continued innovation on jamming and anti-jamming techniques [17]. Constrained attackers, however, are not necessarily less effective, as they can leverage the advanced technologies of SDR, software-defined networking, agile and reconfigurable protocols, sensing, and machine learning. Such capabilities can also provide increases in attack stealth, allowing attackers to avoid detection or localization [18]. Examples of recent energy-conscious attacks include periodic jamming [19]

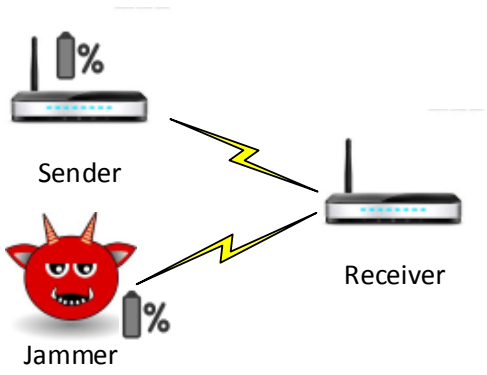


Figure 1.2: We illustrate jamming with three nodes. The defender consists of a transmitter and receiver trying to communicate while the attacker broadcasts signals to disrupt communications. The advent of SDR has enabled new opportunities for research in agile jamming and anti-jamming.

and random jamming [20] that alternate between jamming and sleeping to save energy; control channel jamming [21] and similar attacks that leverage protocol structure for efficiency; and reactive jamming [22], adaptive jamming [18, 23], and mesh jamming [24] which respond to observed activity instead of attacking statically.

Fortunately, the same innovative technologies that enable energy-efficient and stealthy attacks can also enable more robust and agile anti-jamming techniques. The agility provided by SDRs allows defenders and attackers alike to have more fine-grained control of protocols and parameters, enabling the ability to adapt on the fly [25]. However, this mutual agility increases system complexity and presents a significant challenge to our understanding of various performance, security, and reliability metrics required for effective system design.

1.2 Contributions

1.2.1 Insider Attack Against a Known System and Opponent

In Chapter 3, we use optimal control techniques to design an attack against a CPS when the attacker has full-knowledge of the system and the defender's strategy. This scenario occurs during an insider attack when the benign players are using a known control algorithm. Since we know the system and defender's control algorithm we can design an optimization problem to minimize the attacker's input signal constrained to the system's initial state and a desired target state at time T. If this optimization problem is feasible we can solve it to find an optimal attack strategy to drive the system to the desired state at time T. The attacker's desired outcome can include physically destabilizing the system or reducing the system's efficiency. This type of attack may occur in cases of terrorism and corporate sabotage.

We demonstrate the full knowledge attack in a platoon of vehicles using CACC, as described in Section 1.1.1. We assume that one of the cars in the platoon is a malicious inside attacker and desires to cause accidents while not being involved. We show that we can design an attack which causes a final state where the vehicles behind the attacker collide while she is not involved in the collision. Our attacker design uses an optimal set of control inputs to accomplish this attack with minimal effort while maintaining, at least, a minimum safety distance to avoid being involved in a collision. In Chapter 3, we design three different attacks leveraging this optimization approach. In one particularly devastating case, we design attacker that causes the three vehicles behind her to simultaneously collide, sandwiching the middle car, at high-speeds. This causes extensive damage to the three cars involved and could cause a disruption to the highway system that could lead to further accident. We discuss the trade-offs in selecting design parameters because the three attacks we design are only feasible with a limited set of design parameters.

1.2.2 Defending a Critical System Against Unknown Attacks

In Chapter 4, we design a defense for a safety-critical CPS when the system is known but nothing is known about the attacker's strategy or capabilities. We again use the platooning system, introduced in Section 1.1.1, and assume that an inside attacker may exist. We first introduce various attacks and misbehavior to demonstrate the existence and impact of attacks. We show that some attacks can directly cause accidents while other attacks eliminate string stability guarantees.

Since a behaving platoon of vehicles operates in a predictable manner, we propose leveraging the known system and broadcast nature of V2V communications to detect attacks. We design a model using acceleration data from upstream V2V communications so each car can predict the expected behavior of the car directly in front of it. The car can use the predicted behavior compared with radar-based observed behavior to determine if the proceeding car should be trusted. If the observed behavior of the preceding car is similar to the modeled behavior then the detecting car continue to use CACC. If the observed and modeled behavior is very different the detecting car switches to non-cooperative ACC with a longer headway distance. Using attack detection allows us to benefit from the improved performance of CACC while mitigating the impact of attackers.

We simulate various attacks and show their impact on a platoon of cars. For example, in a collision induction attack the attacker claims to accelerate while aggressively decelerating. This attack causes the car following the attacker to accelerate and collide with the attacker, possibly as a high-tech version of insurance fraud. We demonstrate that our detection scheme is fast enough to detect and mitigate this attack avoiding any collisions. We show that our detection strategy is able to mitigate many other attacks with one main exception. Our detection strategy is not able to detect when a greedy player joins without a radar, depending solely on DSRC-based feedforward control.

1.2.3 Interaction of Rational Players with a Known System

In Chapter 5, we consider the interaction of two opponents in a known system who assume their opponent's strategy. We model this using a game theoretic approach, where game theory can loosely be defined as "a bag of analytical tools designed to help us understand the phenomenon that we observe when decision makers interact" [26]. Since both players know the system they are interacting with and assume that their opponent is rational they can find a strategy, or set of strategies, that constitute a Nash equilibrium: a set of strategies for each player such that single-handily deviating does not improve their performance.

We demonstrate this approach using the three node jamming model, introduced in Section 1.1.2, assuming each player has limited energy. The defender and attacker both decide whether or not to transmit and at what power to transmit at. We assume that every action has an energy cost, since even sleeping drains a marginal amount of energy. We design a multi-round 2 player game where players attempt to choose an optimal strategy given both players' initial power levels. We assume that observing the player's opponent is a free action, so the player knows the state of the system (remaining energy level) at all times. This is a reasonable assumption because it simply requires a second radio head and basic processing to determine if and what power they transmit at. The game is then played by each players choosing whether to transmit and at what power level given the current energy levels of both players. We derive an optimal strategy for both player using a dynamic programming approach and extend the game to include frequency hopping options.

Since game theoretic optimallity inherently depends on both players being rational, we compare the performance of rational players with random, constant, and adaptive players. One interesting result that highlights the effect of a non-rational player limiting a rational player is what we call a bullying attack. In this case, an attacker is able to effectively stop a defender by saving its energy and doing nothing, the defender assumes that the attacker is rational though and attacks on each future round. This allows an attacker to be very effective against a rational defender by

simply doing nothing.

1.2.4 Attacking a System with Unknown Parameters

In Chapter 6, we design an attacker who does not know the parameters of the system-under-attack and assumes the defender's strategy is constant. We design this attack using the three-node jamming problem, see Section 1.1.2, with two added assumptions. First, the attacker is battery operated, or resource-constrained, so must conserve its energy to mount the most effective attack. Second, the attacker is used in a military application, so being detected results in a show-of-force.

We develop an adaptive-jamming algorithm [18], where the attacker observes legitimate communications, uses these observations to model her performance, and uses this model to optimize her attack. This three-part technique allows a jammer who does not know the geometry and defender transmit power level to be effective while minimizing energy usage and probability of detection.

We develop two versions of adaptive jamming. The first is a target-based approach where the attacker uses a feedback-controller to cause a target packet delivery ratio (PDR). For example, if the target PDR is 30% the feedback-controller is used to tune the parameters to cause a PDR of 30% with minimum energy usage. The second approach we develop leverages system identification techniques to tune a model of the attacker's impact. We then use this model to select a performance point balancing effectiveness, energy usage, and probability of detection. We implement both of these attacks in software defined radio and demonstrate their effectiveness. We also discuss the tradeoffs for implementing either of the two algorithms.

Chapter 2

Related Work

In this chapter, we introduce related work in the area of cyber-physical security and discuss related work in platooning and jamming.

2.1 Cyber-Physical Security

Popular use of the term cyber-physical systems (CPS) is less than ten years old, exploding after the *2006 NSF Workshop on Cyber-Physical Systems*. At this workshop CPS was described as follows.

> "A cyber-physical system integrates computing, communication and storage capabilities with the monitoring and/or control of entities in the physical world" [27].

Using this definition, a lot of research has previously developed techniques that can be used to design secure and robust cyber-physical systems. We briefly discuss work in the related fields of networked control systems and network security, before discussing recent work on CPS security.

2.1.1 Networked Control Systems

Research in networked control systems (NCS) is traditionally motivated by the need for distributed industrial processes and intra-vehicle automotive communications [28]. NCS continues

to grow in importance with the expansion of CPS and other distributed control applications. A recent survey from Gupta and Chow [28] provides a discussion of historic and recent literature in networked control systems. We introduce some related work in delay compensation, control over lossy networks, and designing secure NCS.

Two major challenges in NCS is how to handle delay and lossy control. The challenge of overcoming delay has been considered in many scenarios including random delays [29, 30], time varying delays [31], how to correct delays [32, 33, 34, 35, 36], stability under delay [37], and control with long delays [38]. Similar work has explored control under lossy networks [39] including TCP-like networks [40] and UDP-like networks [41].

Previous work has also explored security and attack vectors [42] in networked control systems. Amin et. al. explored the design of a controller that is robust to denial-of-service attacks [43]. Mo and Sinopoli [44] discuss a technique to mitigate replay attacks against a networked controller. Spoofing is a major challenge in NCS which can be mitigated using various approaches as described in Gunther's recent survey [45].

2.1.2 Network Security

The field of network security covers a broad range of topics ranging from cloud security to physical-layer wireless security. In this section, we focus on a high level overview of techniques that can be applied to CPS. One classic way to explore security is to consider the CIA triad: confidentiality, integrity, and availability. In wireless CPS all three elements of the CIA model are important. In Section 2.3 we discuss availability of physical layer communications in detail. We refer a reader interested in defending against distributed denial-of-service, or flooding attacks to a a recent survey by Zargar [46].

Integrity is an important network security topic that applies to cyber-physical systems. Integrity involves guaranteeing the origination and timeliness of each packet. Wired networks have traditionally used cryptographic techniques [47] to guarantee the origin of a packet. To

operate effectively these schemes require a trusted third party, pre-shared secrets, or a secret key generation technique. In wireless sensor networks new non-cryptographic techniques have been proposed for authentication [48]. Other techniques have been designed to gurantee packet timeliness to avoid replay attacks [49, 50].

Confidentiality of data is also an important topic in network security. This has been applied to distributed wireless networks through link link layer protocols [51] and network layer protocols [52].

2.1.3 CPS Security

Mo et al. [1] discuss the benefits of using network security and system security techniques to secure cyber-physical systems. The author's then provide compelling arguments that just combining the two security approaches is insufficient for CPSec due to three weaknesses.

1. The system and attack models of both approaches are incomplete for CPSec. For example cyber security does not consider sensor tampering and system security does not consider replay attacks.
2. The security approaches of both approaches are necessary and incomplete in their current form.
3. The countermeasures of both approaches have drawbacks and limitations.

The author's then demonstrate a holistic cyber-physical approach to security to mitigate a network attack, replay, by leveraging system security techniques. This combination of cybersecurity and system security techniques demonstrate the exciting potential of improving security in CPS that this thesis continues.

Other work has explored CPSec in application specific scenarios which allows for accurate assessment of threats and security needs. In particular smart-grid security has received a lot of attention [1, 53, 54]. Another commonly explored CPSec application area, with particular emphasis on privacy, is body area networks for healthcare [55, 56]. In this thesis, we use platooning

and wireless communications as models to explore cyber-physical security. In Section 2.2, we discuss recent research in platooning including relevant platoon security papers. In Section 2.3, we discuss wireless communications as a cyber-physical system focusing on jamming and anti-jamming techniques.

2.2 Platooning

Two approaches are commonly used to analyze formation driving in automotive systems. In microscopic models each car is modeled as a point and their interaction is analyzed while in a macroscopic model the highway system is modeled as a set of pipes with the traffic modeled as a fluid [57]. A comparison of constant time headway policies in the micro and macroscopic models is presented in [58]. Within the domain of platooning, or linear cooperative formation driving, the design criteria most often used is string stability. String stability is roughly defined as the error between vehicles not growing as the length of the platoon grows. In this section, we focus on linear formation driving using a microscopic model.

Various approaches have been suggested for practically implementing string stable platoons. Common design assumptions include the number of radars used and the necessary communication range. For platooning, an engineer can use the assumption from non-cooperative adaptive cruise controls which uses a single front-facing radar [59], implicitly trusting the following car. A design can also use a two radar approach, one rear-facing and one front-facing, for a controller [60] which balances the distance to both the preceding and following car. In this work, we use a controller with a single forward facing radar unit with an implicit trust that the following car will not rear end us.

There are two common assumptions with respect to communication range in the literature. The first is that all cars are able to hear the leader[61], implying the range from the leader to the last car is limited. This assumption forces an artificial bound on platoon size, but this bound fits into many platooning frameworks [62]. The second approach assumes only local communication

from the nearest neighbors [14] and allows for platoons of arbitrary lengths. There have been a few papers that look at optimal control of platoons in a benign case at the micro and macroscopic level[63, 64].

Considerable research has explored the impact of networking on the performance of a string stable platoon of cars. Heemels et al. [13] explore the trade-off between network delay, transmission intervals, and performance on a string stable controller. Tabbara et al. [15] introduce string stability in a platoon of cars with stochastic communications. Zhao et al. [65] explore the effect of stochastic disturbances in the vehicle dynamics and how it impact propagates in a string stable platoon of vehicles. Segata et al. [66] have recently explored the impact of communication performance (fading and transmit power levels) and how communication systems can be designed for platoons [13].

There has been limited research on attacks on platoon controllers or attacks on v2v networks used for platooning. In [2] an attack is designed that decreases the efficiency of a platoon of vehicles. They show they are able to leverage the controller to reduce the efficiency of cars around the attacker by 20-30%. Haas [67] explored an attack on the network of a platooned system showing that jamming, with only a 50% duty cycle, can can cause accidents and platoon deformation. In this work, we continue the exploration of platooning attacks by introducing new attacks using misinformation and malicious controllers.

2.3 Jamming

Due to the potential risk of jamming, a large body of work has recently focused on how to effectively avoid and mitigate the effects of jamming attacks. Much of the work on basic and advanced jamming techniques through the last decade has been summarized in a 2010 survey [17]. Efficient jamming and anti-jamming techniques can be classified into two categories: static and adaptive. Static jamming and anti-jamming techniques rely on specification of protocols, parameters, and strategies in advance, while adaptive techniques rely on context, measurements, and

observations to choose protocols, parameters, and strategies on the fly to improve performance.

2.3.1 Jamming and Anti-Jamming Techniques

One of the simplest forms of jamming is the modulation of a single tone at the carrier frequency, known as tone jamming. Spread spectrum is an effective defense against this basic jamming attack [16], aiming to increase the cost for an attacker to mount an equally-effective jamming attack. One such technique is direct sequence spread spectrum (DSSS) which maps a narrow-band signal to a wider frequency band providing increased robustness of the transmission against a narrow-band attacker through redundancy. A second technique is frequency hopping spread spectrum (FHSS), in which a sender and receiver "hop" between channels using a predetermined schedule. FHSS is very effective at defending against narrow-band attacks provided the two nodes are time-synchronized, the hopping schedule remains secret, and a sufficient number of orthogonal channels are used [68].

Other traditional jamming mitigation techniques have focused on static strategies with shared secrets such as code division multiplexing (CDMA) and orthogonal frequency division multiplexing (OFDM) [69]. Efficient static jamming strategies include random [20], periodic [19], and deceptive jamming [20]. Both random and periodic jamming alternate between attacking and sleeping in an attempt to attack in an efficient manner. Deceptive jamming on the other hand sends legitimate packets in an attempt to stealthily interfere with communications, making its effect very similar to greedy MAC misbehavior techniques [70]. More recent strategies have explored adaptation of protocols and parameters at multiple layers either randomly or in response to observations and measurements. The SPREAD system uses multi-layer adaptation as an extension of spread spectrum [25], providing a more robust communication system but still depending on the same secret-sharing fundamentals. Adaptive jamming strategies using observation-based agility [18] and offline optimization using long-term measurement data [71]. Moreover, adaptive anti-jamming techniques have included the use of advanced signal processing and filtering at the

receiver [72], jamming-aware traffic management [73], and adaptive beamforming [74].

Detection of jamming attacks via system monitoring is another approach to jamming mitigation, allowing the system under attack to change its operation or impose a penalty on the attacker. One such detection technique is to monitor network performance metrics and verify consistency. For example, observing a low packet delivery ratio and consistently high received signal strength, a receiver may infer the presence of a jammer [75]. Such detection techniques can then be used to trigger anti-jamming mechanisms [76, 77, 78].

To counteract the anti-jamming capabilities of spread spectrum, attackers must either increase their resource usage or increase their attack efficiency [75]. An efficient alternative is through random or periodic jamming, in which the jammer alternates between an attacking mode and a sleeping mode to reduce energy usage [79]. Another alternative which combines efficiency and effectiveness is reactive jamming, in which the attacker listens to the channel and transmits a high-power jamming signal when it senses a packet transmission [80]. An additional benefit of random, periodic, and reactive jamming attacks is the reduced likelihood of being detected, a natural protection against the detect-and-respond approaches above.

Another way that attackers can increase efficiency and reduce detectability is by incorporating higher-layer information in the jamming attack formulation. Jamming attackers can incorporate MAC layer information to precisely time jamming emissions [79, 81, 82], for example by jamming the channel when acknowledgement (ACK), clear-to-send (CTS), or data packets are expected according to protocol schedules. Attackers can further incorporate network layer information by observing traffic flows and tuning jammers across the network to minimize network throughput [71].

Attackers can also adapt jamming behaviors based on system performance. Maintaining a network history allows an attacker to decide whether or not it will jam at a particular time using a game theoretic approach [82] or choosing which of a group of jammers should be used at a particular time [71].

2.3.2 Game Theory and Jamming

Game theory has provided a potent tool to investigate and analyze jamming and anti-jamming [83, 84, 85] as well as other security problems. In the domain of jamming, game theory has provided a framework to select parameters and strategies for both static and adaptive jamming and anti-jamming scenarios. We briefly discuss three types of related games: power management games, jammer-versus-defender games, and friendly jamming games.

Power management games study the choice of transmission power levels among nodes in a network to achieve sufficient signal quality while limiting interference with neighbors [86]. Power management games are useful in maximizing the signal-to-interference-and-noise ratio (SINR) of wireless communication in the network. The authors derive a Nash equilibrium for transmission power selection to maximize SINR over the network in both a selfish and cooperative setting.

A second class of relevant games involves explicit competition between jamming and defending players. Previous work has studied the equilibrium behavior of a rate-adaptive defender versus a power-limited jammer [87], choosing jamming power to avoid detection [88], choosing jamming and communication transmission power to balance over-heating concerns [89], choosing jamming strategies considering impact and per-round energy drain [90], and team-versus-team jamming where each team maximizes their own throughput while minimizing the opposing team's throughput [91, 92, 93].

Friendly jamming games aim to use jamming to enforce communication secrecy or privacy against eavesdroppers. In this scenario, utility is defined by the ability to relay data to an intended receiver while preventing eavesdropping by an unintended receiver [94, 95]. Variations on the game include using a coexisting network of active jamming attackers that can also prevent the intended nodes from receiving the data [96].

Our two-player game with energy-constrained players has similarities with many of these related works, but we include the additional consideration of multi-round optimization with a

fixed energy budget for the entire game. The closest of the related works in this regard is the optimal jamming and anti-jamming work of Li et al. [88], but that work differs in that the goal of the attacker is to avoid detection, while in our work the attacker aims to maximally ruin the sender's throughput.

Much work in the stochastic games literature has been focused around iterative algorithms that eventually converge to a Nash equilibrium or approximate Nash equilibrium, posing additional constraints on the game for convergence, such as existence of global optima, or saddle points [97, 98, 99]. For the finite-horizon case, polynomial-time algorithms have been developed, with a running time that is quadratic in the size of the state space [100]. Iterative convergence approaches have also been combined with optimal solving of stage games [101, 102], but without runtime guarantees. Our context allows us to develop a significantly more efficient algorithm that only requires traversing the state space once.

Chapter 3

Insider Attacks Against a Known System and Opponent

In this chapter, we design a full-knowledge attack which leverages deterministic knowledge of both the system-under-attack and the defense strategy. A full-knowledge attack is motivated by inside attackers, where the attacker is a trusted player that is treacherous or compromised. We show that a full-knowledge attacker can have a devastating impact when they leverage their comprehensive knowledge. To demonstrate this attack we use the platooning system discussed in Section 1.1.1 and Section 2.2.

We design an inside attack that causes a high-speed collision in a platoon of cars without the attacker being involved. To design this attack we use a discrete-time optimal control technique [103]. We use a discrete model of the platoon and controller dynamics that are derived assuming that every vehicle, except the attacker, follows a pre-defined control law. The attacker's control inputs are left free during a set time period that we call the attack horizon.

We use the discrete platoon model to design a constrained quadratic optimization problem that minimizes the attacker control inputs and whole system's state deviance over the attack horizon. We force the system to behave realistically by using various constraints for the optimization. First, we use the system update equations as a constraint for how the system behaves. Second,

we limit the vehicle behavior using realistic boundaries. For example, we set a maximum acceleration since (most) cars are not able accelerate at $500\frac{m}{s^2}$. We also use constraints to select the attack behavior. The attack is designed with an equality constraint to force the last state of the optimization problem to a desired attack state. We use a similar constraint over the entire attack horizon to avoid causing other, unintended, collisions. This constrained quadratic optimization problem can then be solved using an off-the-shelf optimization software like the Gurobi optimization package [104]. We discuss the trade-offs in selecting constraints and what are feasible attack subsets in Section 3.3.

We demonstrate three attacks using this optimization approach. The follower attack is mounted by the third car of a five car platoon and causes the fourth and fifth car to collide, while the attacker drive away safely. The distant follower attack is similar but the third car in a seven car platoon causes the sixth and seventh car to collide. An attacker mounting the distant follower attack has the advantage of a larger safety distance. Lastly, in the sandwich attack a malicious car causes the three cars following her to simultaneously collide. In all of these attacks, the attacker is able to accomplish their goal without being involved in the accident. These attacks in a semi-automated highway system not only causes major damage, but would cause a non-linear chaotic event that would lead to unpredictability for the other highway users.

To summarize, we make the following contributions in this chapter.

- We introduce an optimization problem that allows for the design of novel inside attacks against platoons of cars.
- We demonstrate this technique by designing three optimal attacks that cause vehicles following the attacker to collide while the attacker is not involved in the collision.
- We discuss the trade-offs in choosing design parameters in these attacks.

The remainder of this chapter is organized as follows. We formulate the platooning system in Section 3.1 and discuss our attack formulation in Section 3.2. We design three attacks and simulate them in Section 3.3. Lastly, in Section 3.4, we discuss limitations and future directions.

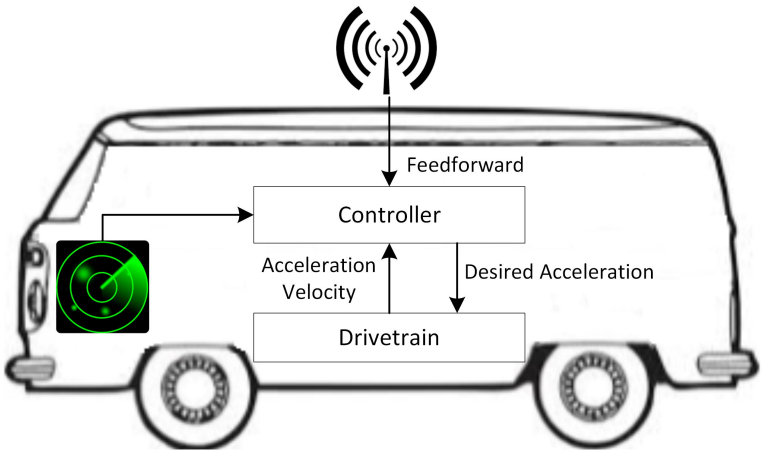


Figure 3.1: In this figure, we show our controller structure for a platooned vehicle. The vehicle uses radar to determine distance and error from the car in front of it, DSRC to get feedforward information from the proceeding car, and powertrain measurements to determine its current state.

3.1 System Model

Our system consists of a platoon of K cars that that we number from 0 to $K-1$ with car 0 being the platoon's leader. We assume that the cars all drive in a single straight lane and that their order can not change. We indicate the spatial position, velocity, and acceleration of car i as q_i, v_i, and a_i respectively. Without loss of generality, we assume that the car at the position, q_i, has zero width so we do not define separate rear and front vehicle locations. We indicate the distance between car i and car $i-1$ as $d_i = q_{i-1} - q_i$ with $d_0 = 0$ and the desired distance between car i and car $i-1$ as $d_{r,i}$ with $d_{r,0} = 0$. We define the error for car i as $e_i = d_i - d_{r,i}$.

The cars desire to follow a constant headway policy such that $d_{r,i} = h_{d,i}v_i + L_i$ where L_i is a constant distance offset and $h_{d,i}$ is the the desired headway of car i. We can substitute the constant headway policy into our error equations to get

$$e_i = q_{i-1} - q_i - h_{d,i}v_i - L_i. \tag{3.1}$$

We can set the distance $L_i = 0$ in (3.1) by assuming a change of basis to provide for the safe stopped distance such that $e_i = q_{i-1} - q_i - h_{d,i}v_i$. The choice of L_i must be considered when

designing an attack or defense in platoons. Attack design must be adjusted to the actual stopped distance.

We model the cars using the common double integrator model [60] with a lag constant of η_i for each car. Given a desired acceleration of u_i, car i has the following continuous time differential equations.

$$\dot{a}_i = -\eta_i^{-1} a_i + \eta_i^{-1} u_i \tag{3.2}$$

$$\dot{v}_i = a_i \tag{3.3}$$

$$\dot{q} = v_i \tag{3.4}$$

$$\dot{e}_i = v_{i-1} - v_i - h_{d,i} a_i. \tag{3.5}$$

3.1.1 Controller

In Figure 3.1 we show a vehicle that is equipped for cooperative adaptive cruise control. We assume a radar plus DSRC setup and use a controller that has been tested for this setup [13]. This controller uses a combination of a DSRC based feedforward input, $u_{ff,i}$, and a measurement based feedback input, $u_{fb,i}$, such that

$$u_i = u_{fb,i} + u_{ff,i}. \tag{3.6}$$

The feedforward input is used to predicatively tune the control signals based on what is expected to happen while the feedback input is used to respond to measured error. The inter-vehicle distance is measured using radar and used to calculate error which allows for proportionate-derivative feedback control such that

$$u_{fb,i} = k_p e_i + k_d \dot{e}_i. \tag{3.7}$$

The feedforward controller is provided via DSRC using the update equation

$$\dot{u}_{ff,i} = -h_{d,i}^{-1} u_{ff,i} + h_{d,i}^{-1} \hat{u}_{i-1}, \tag{3.8}$$

where $\hat{u}_{i-1}$ is received via DSRC. In the case that all vehicles are behaving then $\hat{u}_{i-1} = u_{i-1}$ during update periods. However, in general, we assume this equation may not hold in order to account for malicious behavior. A simulation of 5 cars using this controller to accelerate and decelerate simultaneous is provided at `http://goo.gl/zqMtpU`.

It is important to note that car 0 has a unique control law such that $u_0 = u_r$ where u_r is a reference desired acceleration profile. It is assumed that car 0 is given u_r in real time so no non-causal predictions can be made. The proposed controller has been shown to be string stable in continuous communication systems and has been tested in real networked platoons with delays and sampling [59].

3.1.2 System Description

We define the vector $x_i^T = [e_i, v_i, a_i, u_{ff,i}]$ for the state of car i. The update equation for a vehicle can be written as a linear system such that

$$\dot{x_i} = A_{i,i} x_i + A_{i,i-1} x_{i-1} + B_{s,i} u_i + B_{c,i} \hat{u}_{i-1}, \; \forall \, i > 0 \tag{3.9}$$

and

$$\dot{x_0} = A_0 x_0 + B_{s,i} u_r \tag{3.10}$$

where

$$A_{i,i} = \begin{pmatrix} 0 & -1 & -h_{d,i} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\eta_i^{-1} & 0 \\ 0 & 0 & 0 & -h_{d,i}^{-1} \end{pmatrix}, \tag{3.11}$$

$$A_{i,i-1} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3.12}$$

$$B_{s,i}^T = \begin{pmatrix} 0 & 0 & \eta_i^{-1} & 0 \end{pmatrix}, \tag{3.13}$$

$$B_{c,i}^T = \begin{pmatrix} 0 & 0 & 0 & h_{d,i}^{-1} \end{pmatrix}, \tag{3.14}$$

and

$$A_0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & -\eta_0^{-1} & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{3.15}$$

We define X as the state of the whole systems so that $X^T = [x_0^T, x_1^T, ...x_{K-1}^T]$. We define the inputs to the system as $U^T = [u_0, \hat{u}_0, u_1, \hat{u}_1, \ldots, u_{K-1}]$ where each vehicle chooses its input values u_i and $\hat{u}_i$. This allows us to write the linear equations for the whole system as

$$\dot{X} = AX + BU \tag{3.16}$$

where

$$A = \begin{pmatrix} A_0 & 0 & 0 & \ldots & 0 & 0 \\ A_{i,i-1} & A_{i,i} & 0 & \ldots & 0 & 0 \\ 0 & A_{i,i-1} & A_{i,i} & \ldots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & A_{i,i-1} & A_{i,i} \end{pmatrix} \tag{3.17}$$

and

$$B = \begin{pmatrix} B_{s,i} & 0 & 0 & \ldots & 0 & 0 \\ 0 & B_{c,i} & B_{s,i} & \ldots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & B_{c,i} & B_{s,i} \end{pmatrix}. \tag{3.18}$$

3.1.3 General Discrete Attack

We consider a scenario in which the third car in the platoon mounts the attack and the other cars follow the prescribed control algorithm with $U_r = 0$ such that the platoon is at a stable speed and not changing acceleration, except for the attack. Thus during the attack $U = [0, 0, 0, 0, u_a, \hat{u}_a, u_{fb,3}+u_{ff,3}, u_{fb,3}+u_{ff,3}, u_{fb,4}+u_{ff,3}, u_{fb,3}+u_{ff,4}...u_{fb,N-2}+u_{ff,N-2}, u_{fb,N-1}+u_{ff,N-2}, u_{fb,N-1}+u_{ff,N-1}]$. We define two new control vectors, the attack control vector and the normal control vector defined as $U_a = [u_a, \hat{u}_a]$ and $U_n = [u_{fb,3} + u_{ff,3}, u_{fb,3} + u_{ff,3}, u_{fb,4} + u_{ff,3}, u_{fb,3} + u_{ff,4}...u_{fb,N-2} + u_{ff,N-2}, u_{fb,N-1} + u_{ff,N-2}, u_{fb,N-1} + u_{ff,N-1}]$ respectively.

We can then define our update equation as

$$\dot{X} = AX + B_aU_a + B_nU_n = AX + B_aU_a + B_nKX = (A + B_nK)X + B_aU_a \tag{3.19}$$

since U_n can be wrote as a linear combination of X such that $U_n = KX$ using (3.6). We define $A_a = (A + B_nK)$ so

$$\dot{X} = A_aX + B_aU_a. \tag{3.20}$$

We discretize this equation using a sampling time of 100 ms to arrive at

$$X[n + 1] = A_{d,a}X[n] + B_{d,a}U_a[N]. \tag{3.21}$$

3.2 Attack Optimization

We design an attacker that minimizes the state deviance and input effort. To do this we define a quadratic cost function

$$\begin{aligned} J[X, U_a] =& (X[N] - X_r)' Q (X[N] - X_r) + \\ & \sum_{i=1}^{N-1} (X[i] - X_r)' Q (X[i] - X_r) + U_a[i]' R U_a[i] \end{aligned} \tag{3.22}$$

where X_r is the desired state of the system when no attack is occurring which we assume is equal to $X[1]$.

We use (3.22) to define our attack optimization problem as

$$\begin{aligned} \underset{U_a}{\text{minimize}} \quad & J[X, U_a] \\ \text{subject to} \quad & X[i+1] = A_{d,a} X[i] + B_{d,a} U_a[i] \; \forall \; i \in [1, N-1] \\ & L_b \leq C_1 X[i] \leq G_b \; \forall \; i \in [1, N] \\ & \psi_d = C_2 X[N] \\ & X[1] = X_0 \end{aligned} \tag{3.23}$$

where L_b and G_b are the lower and upper bounds for linear combinations of states defined using C_1. Similarly $\psi_d = C_2 X$ is an equality constraint defining the attack goal, where the attacker chooses C_2 and ψ_d. The constraints must be carefully selected to make sure they are feasible. Constrained quadratic optimization problem can be solved by using any number of tools, we use the commercial Gurobi solver [104] to leverage its efficient algorithms.

In Section 3.3, we use (3.23) to design three attacks: the follower attack, the distant follower attack, and the sandwich attack.

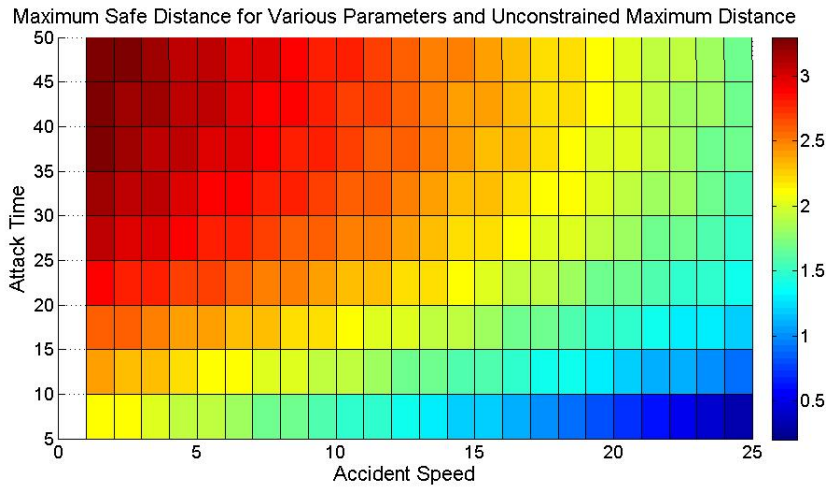


Figure 3.2: In this figure, we show the front between maximum safety distance, attack time, and accident speed at collision when the attacker does not have a bound on their maximum following distance.

3.3 Attacks and Simulations

In this section, we design three attacks using the attack formulation (3.23) and the Gurobi optimizer. We discuss each of these attacks and for the full knowledge attack highlight the trade-offs in constraint selection.

3.3.1 Follower Attack

In the follower attack, the attacker causes a collision between the two cars directly following her at time T while not being involved in the collision. We define ψ_d such that the distance between car $a+1$ and $a+2$ is zero at time N and their velocity is at a threshold v_a. For all time steps we define upper and lower bounds on the distance between the attacking car and other cars which we call the attacker's safety distance. We also constrain all car velocities and accelerations using realistic bounds.

We found that the parameters, v_a, safety distance, and attack time must be chosen carefully to allow for a feasible solution. In Figure 3.2 we show the trade-off between safety distance, accident time, and accident speed a five car platoon with the third car attacking. It is interesting

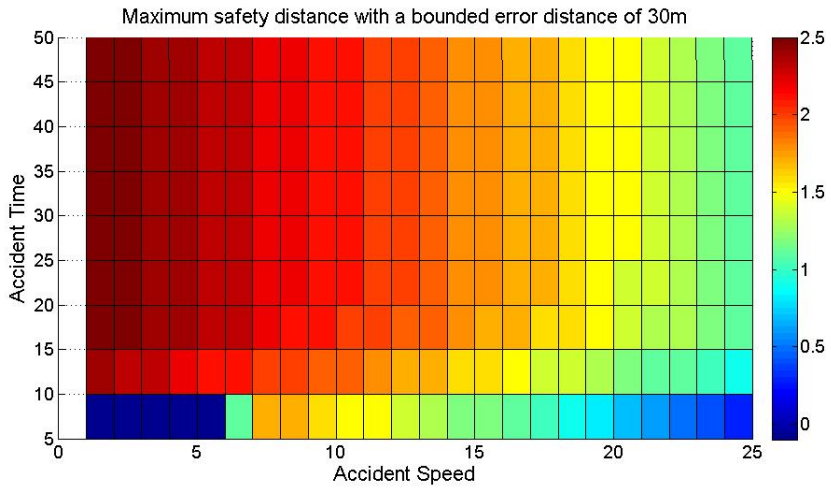


Figure 3.3: In this figure, we show the front between maximum safety distance, attack time, and accident speed at collision when the attacker has a bound of 30m of seperation from the platoon.

to note, with many of these parameter combinations the attacker falls behind from the second car, effectively creating a second platoon. To limit this effect we also define a constraint on the attacker's maximum separation from the platoon. In Figure 3.3 we show the trade-off curves, but now set a maximum separation distance. In both of these figures, it is clear that the maximum attacker safety distance decreases as the attack speed increases. Similarly, their is some increase with accident time but this effect is less pronounced.

We simulate the follower attack mounted by the third car in a five car platoon in Figure 3.4. A video of the attack simulation can be found at `http://goo.gl/YdpcaZ`. In this figure, we show that the attacker causes a high speed accident without being involved. The attacker does get close to the accident, at some points within 2.5 meters, but is never involved.

3.3.2 Distant follower attack

In the distant follower attack, the attacker causes a collision between two cars following her at time T while not being involved in the collision. The attack is differentiated by the colliding cars being further back in the platoon, which allows the attacker to increase her safety distance.

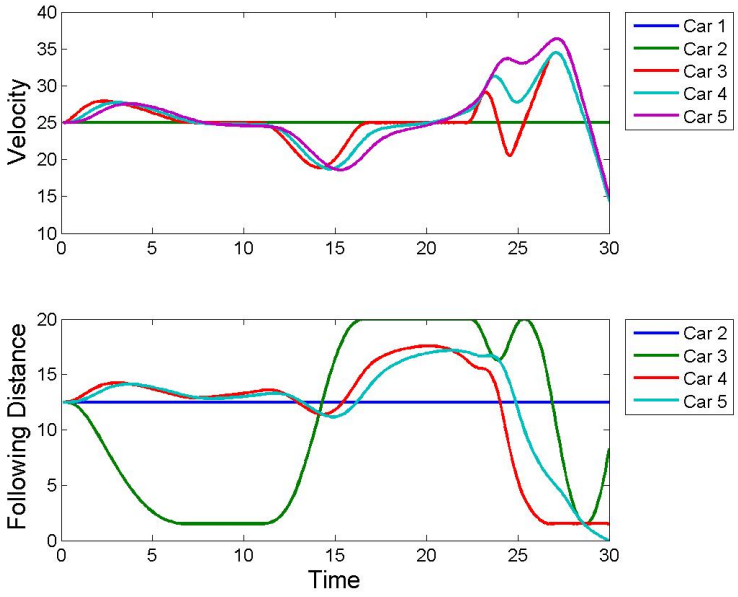


Figure 3.4: In this figure, we show the results for a follower attack moutned by the third car in a 5 car platoon. The attack causes car 4 and 5 to collide at 30 seconds with a velocity of 15 $\frac{m}{s}$. The attacker never gets closer than 2.5 meters to any other car in the platoon. A video of this attack simulation is found at `http://goo.gl/YdpcaZ`.

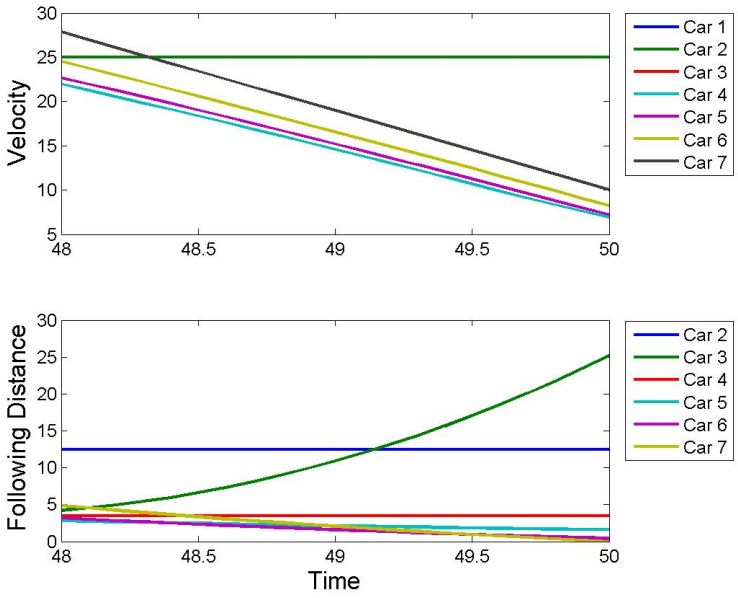


Figure 3.5: In this figure, we show the last two seconds of the attack against a seven car platoon.

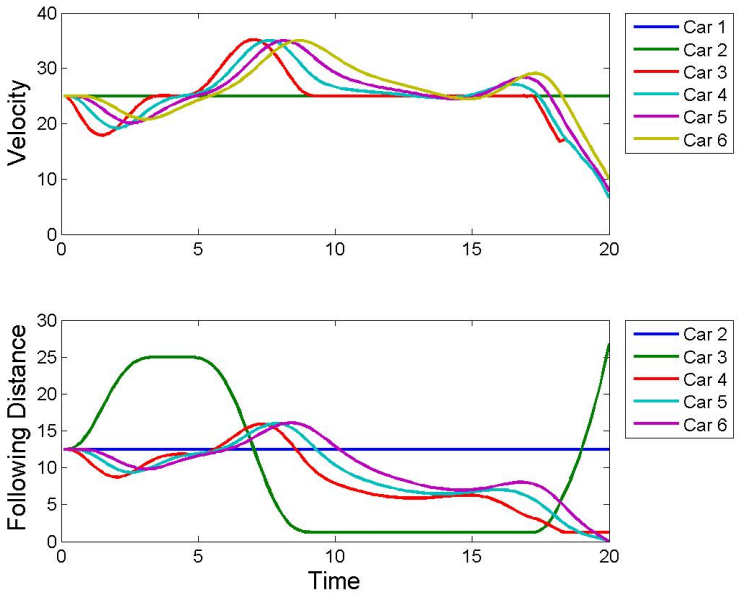


Figure 3.6: We demonstrate an attack in a 6 car system where the third car causes a simultaneous double collision between cars 4, 5, and 6. A video of this simulation can be found at `http://goo.gl/kFFVNZ`

Similar to the follower attack parameters must be selected considering the tradeoff in their performance. We define one more parameter, the non-attacker safety distance, which is used to bound how close a car that is not targeted gets to the accident. Using a low non-attacker safety distance, increasing the risk for other cars not under attack, allows us to define a higher attacker safety distance. An increased attacker safety distance decreases the risk to the attacker even further. In Figure 3.5, we show the last two seconds of a distant follower attack mounted by the third car in a seven car platoon. In this case, the sixth and seventh car collide at $10\frac{m}{s}$ while the attacker never gets closer then 3 meters to any other vehicle.

3.3.3 Sandwich attack

In the sandwich attack, the attacker causes three cars to simultaneously collide, with similar trade-offs for selecting parameters as discussed in the previous two attacks. In Figure 3.6, we simulate this attack mounted by the third car which causes the fourth, fifth, and sixth car to

simultaneously collide. This attack damages all the cars, but car 5 would be especially injured. A video of this simulation can be found at `http://goo.gl/kFFVNZ`.

3.4 Discussion

In this chapter, we design an attack that leverags knowledge of the system-under-attack and the defender's strategy. We formulate a general optimization problem that an attack designer can use to select the desired system state at the end of an attack horizon. We demonstrate the use of this optimization problem with three example attacks. In one particularly devastating attack, the sandwich attack, we design an attack that causes the three cars following her to simultaneously collide while she safely drives away.

3.4.1 Limitations and Future Work

We can expand this work in multiple directions. First, we did not consider the stochastic nature of vehicles on a real highway. So in an actual highway scenario this attack may not be as effective or safe (for the attacker). We propose using a state-based discrete optimal controller to improve this attack and allow it to be robust to noise. To do this an attack can be designed such that the attacker's chosen control inputs are selected based on the state-based optimal strategy.

Second, in our attack design constraint parameters must be carefully selected to guarantee a feasible solution. It would be interesting to formalize and visualize the trade-offs in design parameter selection. This can allow attacks to quickly be designed in a practical scenario allowing for state-based optimal strategies.

Chapter 4

Defending a Known Critical System Against an Unknown Attacker

In Chapter 3, Chapter 5, and Chapter 6 we assume that the player has full-knowledge of her opponent's strategy. In this chapter, we design a defense strategy for a critical system that makes no assumption of the attacker's strategy. To do this we assume the defender has two operating modes: collaborative and safe. The defender defaults to collaborative but monitors other player's behavior to detect abnormalities. If abnormal behavior is detected the defender switch to safe mode. We demonstrate this technique in a platoon of cars as discussed in Section 1.1.1, Section 2.2, and Chapter 3.

In this chapter, we explore what happens when one of the cars in the platoon does not behave according to the control law. Such a vehicle can be malicious, greedy, or even a malfunctioning benign vehicle. We are particularly interested in an insider attack where the attacker either uses a malignant control law or misreports information about her behavior. We introduce a set of 5 different attacks and abnormal behaviors and briefly discuss their motivation and impact on the system. One particularly devastating attack is the collision induction attack where the attacker broadcasts that she is accelerating while in reality, the attacker jams on her brakes. This attack causes the preceding car to collide at high speeds and, with high probability, results in loss of life

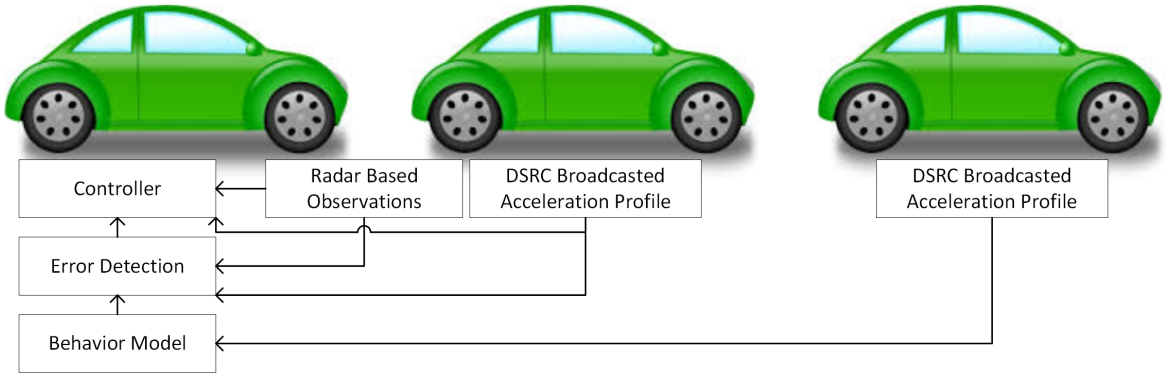


Figure 4.1: In this figure, we show our proposed detection scheme at a high-level. The car in the back of the platoon uses data sent via DSRC by the first car to model the expected behavior of car 2. The car then determines whether the two expected signals differ by an amount greater than a threshold.

and assets.

Given the existence of misbehavior and attacks that can be mounted in a platoon of vehicles we propose using a model based detection scheme to detect and mitigate the impact of malicious behaviors. We propose each vehicle model the expected behavior of the vehicle proceeding them using DSRC information provided from cars farther up the platoon. Vehicles can use this model to calculate the error between the modeled states and the measured state of the proceeding car. The error calculations can then be used with a simple threshold to calculate whether an abnormality exists in the system or not. The technique for modeling, calculating error, and thresholding can all be chosen in the design of the system. We summarize this approach in Figure 4.1.

Once an attack is detected we propose that the vehicle changes to a non-cooperative ACC protocol with an increased headway distance to guarantee safe performance. We are able to detect most abnormalities and are able to avoid the collision that would be caused by the collision induction attack using this technique. This detection scheme could be combined with a global reputation system to keep track of whether or not certain vehicles are often problematic.

To summarize, in this chapter we make the following contributions.

- We propose a set of insider attacks that can cause unexpected behavior in platoons and

may cause fatal accidents.

- We develop a platoon detection method based on up stream DSRC communications to detect misbehavior.
- We design a two state operating mode for semi-autonomous cars to safely transition to a non-cooperative cruise control when attacks are being mounted.
- We simulate the above attacks, detection, and mitigation schemes to provide a proof-of-concept.

The rest of this chapter is organized as follows. In Section 4.1 we introduce our system models. In Section 4.2 we introduce our system attacks along with their motivation and impact. In Section 4.3 we introduce our detection scheme. In Section 4.4 we introduce our simulation and the results of our detection scheme. In Section 4.5 we discuss the trade-offs and possible improvements in the detection system design as well as future work. We present a discussion of related work in platooning in Section 2.2

4.1 System Model

In this chapter, we use the model and assumption from Section 3.1 through Section 3.1.2.

We discretize the system given in (3.16) since the controller is implemented on a digital computer using digital communications. We assume the radar has a sampling time of 1 ms and the communication system has a sampling time of 100 ms. We assume that the controller uses a sample and hold technique for the communication input variable. We thus use the update equations

$$X[k+1] = A_d X[k] + B_d U[k] \tag{4.1}$$

where A_d and B_d represent an exact discretized version of (3.16).

For cars that follow the control law we can similarly define the controller equations for car i

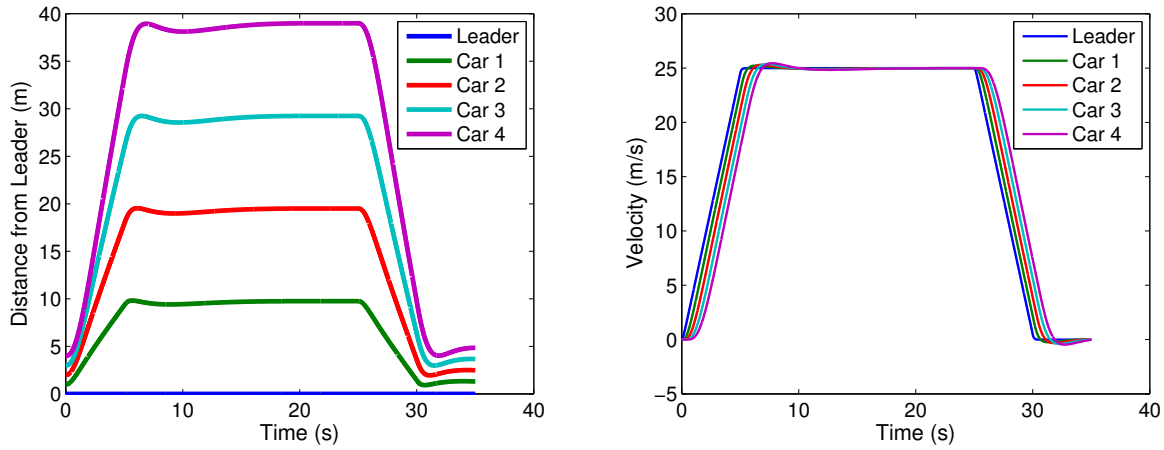


Figure 4.2: In this figure, we show a simulation of our system without attacks. On the left, we show a plot the distance for each car behind the lead car. On the right, we show the velocity for each of the cars. A video simulating this system is provided at `http://goo.gl/zqMtpU`.

in term of x_i and x_{i-1} as

$$u_i = k_1 x_i[k] + k_2 x_i[k-1] \tag{4.2}$$

where

$$k_1 = \begin{pmatrix} k_p + \frac{k_d}{.001} & 0 & 0 & 1 \end{pmatrix} \tag{4.3}$$

and

$$k_2 = \begin{pmatrix} -\frac{k_d}{.001} & 0 & 0 & 0 \end{pmatrix}. \tag{4.4}$$

The input $u_i[k]$ is updated every 1 ms while $\hat{u}_i[k]$ is updated every 100 ms and kept constant otherwise. We model this system for a platoon of 5 cars in Figure 4.2 where the platoon accelerates for 5 seconds, holds for 15 seconds and then decelerates for 5 seconds. The controller in this figure uses a CACC controller with a constant headway of .35 seconds and constant spacing of 1 m.

We assume homogeneous cars such that $A_{i,i-1}$, $A_{i,i}$, $B_{s,i}$, and $B_{c,i}$ are known and the same

for all vehicles. This allows us to write our discrete matrices as

$$A_d = \begin{pmatrix} A_{d,0} & 0 & 0 & \dots & 0 & 0 \\ A_{d,1} & A_{d,2} & 0 & \dots & 0 & 0 \\ 0 & A_{d,1} & A_{d,2} & \dots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \dots & A_{d,1} & A_{d,2} \end{pmatrix}$$

where $A_{d,0} \in \mathbb{R}^{4x4}$, $A_{d,1} \in \mathbb{R}^{4x4}$, and $A_{d,2} \in \mathbb{R}^{4x4}$. Likewise we define

$$B_d = \begin{pmatrix} B_{d,0} & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ B_{d,1} & B_{d,2} & B_{d,3} & \dots & 0 & 0 & 0 & 0 \\ \vdots & & & \ddots & & & & \vdots \\ 0 & 0 & 0 & \dots & B_{d,1} & B_{d,2} & B_{d,3} & 0 \end{pmatrix}$$

where $B_{d,0} \in \mathcal{R}^{4x1}$, $B_{d,1} \in \mathcal{R}^{4x1}$, $B_{d,2} \in \mathcal{R}^{4x1}$, and $B_{d,3} \in \mathcal{R}^{4x1}$.

Attack	Impact	Motivation	Method
Reduced Headway Attack	Decreased String Stability	Decreased fuel consumption Increased density	Misbehavior
Joining Without Radar	Decreased String Stability	Decreased cost over radar equipped car	Misbehavior
Mis-report Attack	Decreased Performance	Mistrust of the system	Misinformation
Collision Induction Attack	Collision Loss of Life Property Damage	Maliciousness Terror	Misbehavior & Misinformation
Non-Attack Abnormalities	Decreased Performance Decreased String Stability	Improper Maintenance	Misbehavior

Table 4.1: In this table, we summarize the system level attacks that we propose including their impact, method, and motivations.

4.2 Attack Strategies

In this section, we introduce a set of attacks and abnormal behaviors that can occur in a platooned vehicular network. We discuss possible motivations for the attacks which range from rational to byzantine. This list is not comprehensive but provides a start to the discussion of what system

level attacks may impact a formation of vehicles and how serious the effects of these attacks are. For the convenience of the reader we summarize these attacks, their motivation, their potential impact, and their implementation in Table 4.1.

In the remainder of this section we refer to the attacker's control input and performance parameters using the letter 'a' in the subscript. Thus, u_a refers to the attacker's control signal during the attack and $\hat{u}_a$ refers to the attacker's broadcasted control signal. We assume that the attacker's signal is non-additive so the state update equation for the attacker become $x_a[k+1] = Ax_a[k] + Bu_a[k]$.

4.2.1 Reduced Headway Attack

In the current highway system the majority of motorists do not follow the recommended 2 second headway speed, with many studies showing the average speed on freeways being under 1 second [10]. This attack models a similar greedy behavior where a car ignores the recommended headway speed that guarantees string stability and follows closer. A driver might, for example, follow at a headway of 0.125 second speed when the vehicles in the platoon are only string stable at headway distance greater than or equal to a 0.25 second. This attack would be implemented by a driver who wants to increase fuel savings by decreasing draft or a driver who manually drives with extremely small headways.

To implement this attack we change the attacker's headway parameter to $h_{d,a} < h_{d,min}$ where $h_{d,min}$ is the recommended minimum headway speed.

4.2.2 Joining Without Radar

This is another greedy behavior where a car attempts to become part of a platoon without having the necessary radar, or other distancing equipment. This is motivated by a driver who does not want to buy a new vehicle but retrofits a car with DSRC which, unlike radar, does not require per vehicle tuning. This attack causes the reaction of the car to be based only on the feedforward

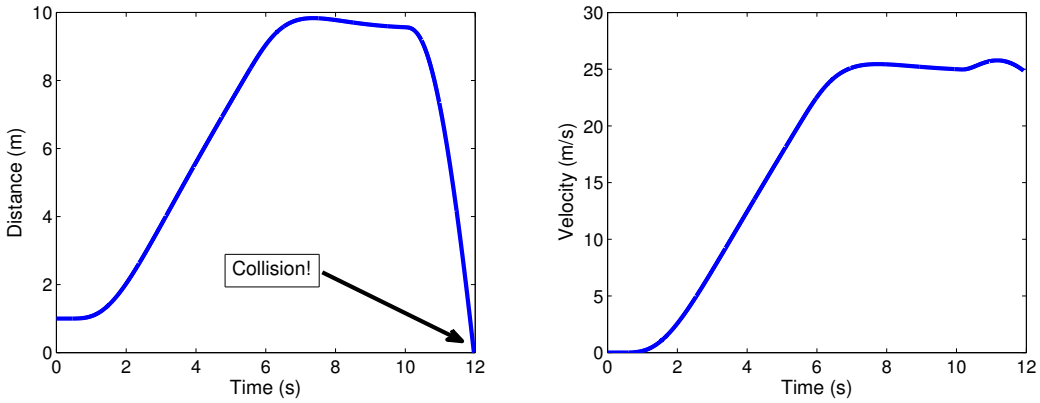


Figure 4.3: In this figure, we show the effect of a collision induction attack that is started at 10 seconds. On the left is a plot of the distance between the attacker and the car under attack and on the right is a graph of the car under attacks velocity. In under 2 seconds the attacking car is hit by the following car going at a speed of over 55 miles per hour. We provide a video simulating this system at `http://goo.gl/IVvIcP`

information which is dangerous if wireless congestion prevents the cars from communicating properly. This also eliminates the guarantees that are provided by string stability to the platoon of cars, increasing the risk of an accident.

This attack is implemented by changing the attacker's control law to $u_a = u_{ff,a}$ and ignoring the feedback portion of the control law.

4.2.3 Mis-report Attack

This is an attack that can be mounted for various reasons including not trusting the cooperative adaptive cruise control system. The attacker misinforms the vehicle that is following to increase the following car's headway or to cause a change in the following car's behavior. The attacker mounting this attack can either follow the prescribed control law or choose an alternative control law. In this work, we assume that the attacker follows the prescribed control law and only misreports its behavior so $u_a = u_i$. This attack is motivated by wanting to increase the following distance of the preceding car.

The attacker defines a mis-report percentage $\beta \in [0, 1]$ and then implements the attack by reporting $\hat{u}_a = (1 - \beta)u_a$ if $u_a > 0$ and $\hat{u}_a = (1 + \beta)u_a$ if $u_a < 0$.

4.2.4 Collision Induction Attack

In this attack, the attacker broadcasts an acceleration profile indicating that they are speeding up which causes the following vehicle to accelerate. The attacker actually starts to aggressively brake which causes the error between the attacker and following car to quickly increase. This is very likely to cause an accident at high speed which makes this attack extremely dangerous.

This is very similar to attacks that could be mounted in the current highway system. If a driver was to jam on their breaks during rush hour while being tailgated, the vehicle would be rear ended. If there are many cars that are all tailgating this can result in a multi-car pile up. In Figure 4.3 we show this attack implemented starting at ten seconds. We provide a video simulation of this attack at `http://goo.gl/IVvIcP`. In under two seconds the car behind the attacking car collides with the attacker at speeds over 25 meters per second, or approximately 56 miles per hour.

Assuming that cars have a range on their inputs defined as $u_i \in [u_{min}, u_{max}]$ we can implement this attack by setting the attackers control parameters to $u_a = u_{min}$ and $\hat{u}_a = u_{max}$.

4.2.5 Non-attack abnormalities

Our detection method is also able to detect non-malicious abnormal behaviors in the system. For example, our detection scheme detects if the acceleration or breaking parameters of a vehicle were to change due to normal wear on the system. This can be used in conjunction with a global monitoring system to help alert drivers when their vehicle might need maintenance.

To model abnormal driving in our system we vary the value of η_a for a vehicle that we call the attacker even though their intent may not be malicious.

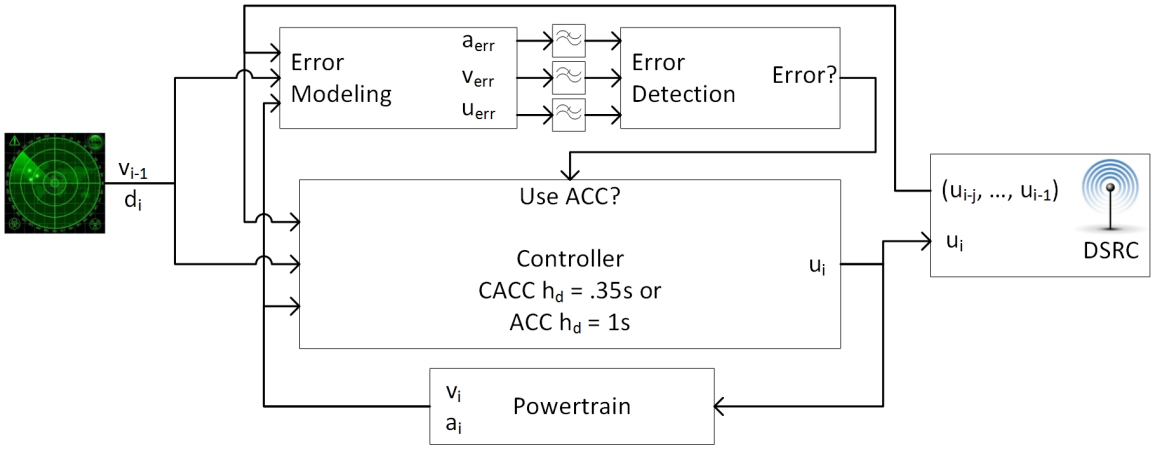


Figure 4.4: In this figure, we show a detailed diagram of our proposed detection scheme. A model of the expected behavior of the car in front of the monitoring car is made from the broadcasted upstream control information. This is compared to the measured behavior of the car in front of the monitoring car. If the error is larger than expected, the monitoring car switches to a non-cooperative ACC algorithm.

4.3 Model Based Attack Detection

In Figure 4.4, we show our proposed approach to detecting abnormal behavior in a platoon of cars. Our approach has every car model the expected behavior of the vehicle directly in front of them. The vehicles then compare the calculated expected behavior with the observed behavior. Using these comparisons the car is then able to detect both malicious and benign abnormalities. The ability to detect malicious as well as benign but dangerous behavior is one of the greatest strengths of our approach.

Once abnormal behavior is detected, the car switches from operating in a cooperative platoon framework to a radar only based adaptive cruise control framework where it is safe even if the preceding car is mounting an attack. We choose this very aggressive response to a potential attack for a multiple reasons. First, the potential impact of a malicious car is a high-speed traffic accident which, in the worst case, results in loss of life, and, in the best case, results in high-value property damage. This technique can also be combined with regional reputation systems to detect vehicles that frequently behave abnormally.

We discuss each portion of our detection and response scheme in detail below.

4.3.1 Modeling Techniques

In this section, we design an algorithm for car i to model the expected behavior of car $i-1$ given the data packets from car $i-j$. We define $x_{m,i-1}$ as the modeled state of car $i-1$. We can then define the state of all the cars in the model as $X_m = [x_{m,i-j}, x_{m,i-j+1}, \ldots, x_{m,i-1}]$. Likewise, we define the feedback inputs and feedforward inputs of car $i-1$ as $u_{m,i-1}$ and $\hat{u}_{m,i-1}$ respectively. This allows us to define the inputs at each time as.

$$U_m = [u_{m,i-j}, \hat{u}_{m,i-j}, u_{m,i-j+1}, \hat{u}_{m,i-j+1}, \ldots, u_{m,i-1},]^T.$$

We can write the system update equation for the model as

$$X_m[k+1] = A_m X_m[k] + B_m U_m[k] \tag{4.5}$$

where

$$A_m = \begin{pmatrix} A_{d,0} & 0 & 0 & \ldots & 0 & 0 \\ A_{d,1} & A_{d,2} & 0 & \ldots & 0 & 0 \\ 0 & A_{d,1} & A_{d,2} & \ldots & 0 & 0 \\ \vdots & & & \ddots & & \vdots \\ 0 & 0 & 0 & \ldots & A_{d,1} & A_{d,2} \end{pmatrix} \tag{4.6}$$

and

$$B_m = \begin{pmatrix} B_{d,0} & 0 & 0 & \ldots & 0 & 0 & 0 \\ B_{d,1} & B_{d,2} & B_{d,3} & \ldots & 0 & 0 & 0 \\ \vdots & & & \ddots & & & \vdots \\ 0 & 0 & 0 & \ldots & B_{d,1} & B_{d,2} & B_{d,3} \end{pmatrix}.$$

We assume that the cars in the model behave according the control law given in (4.2) with 1ms radar update times and 100ms state broadcast times.

During an update period we can use (4.2) to define

$$U_m[k] = \phi_1 X_m[k] + \phi_2 X_m[k-1] + \phi_3 \hat{u}_{i-j}[k] \tag{4.7}$$

where

$$\phi_1 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_1 & 0 & \dots & 0 \\ 0 & k_1 & 0 & \dots & 0 \\ 0 & 0 & k_1 & \dots & 0 \\ 0 & 0 & k_1 & \dots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & 0 & 0 & \dots & k_1 \\ 0 & 0 & 0 & \dots & k_1 \end{pmatrix}, \phi_2 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & \dots & 0 \\ 0 & k_2 & 0 & \dots & 0 \\ 0 & 0 & k_2 & \dots & 0 \\ 0 & 0 & k_2 & \dots & 0 \\ \vdots & & & \ddots & 0 \\ 0 & 0 & 0 & \dots & k_2 \\ 0 & 0 & 0 & \dots & k_2 \end{pmatrix}$$

and

$$\phi_3 = \begin{pmatrix} 1, 1, 0, \dots, 0 \end{pmatrix}^T. \tag{4.8}$$

Likewise, during a non-update period we can define our update input as

$$U_m[k] = \phi_4 X_m[k] + \phi_5 X_m[k-1] + \phi_6 U_m[k-1] \tag{4.9}$$

where

$$\phi_4 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & k_1 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & k_1 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix}, \phi_5 = \begin{pmatrix} 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & k_2 & 0 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & k_2 & \dots & 0 \\ 0 & 0 & 0 & \dots & 0 \\ \vdots & & & \ddots & \vdots \\ 0 & 0 & 0 & \dots & k_2 \\ 0 & 0 & 0 & \dots & 0 \end{pmatrix},$$

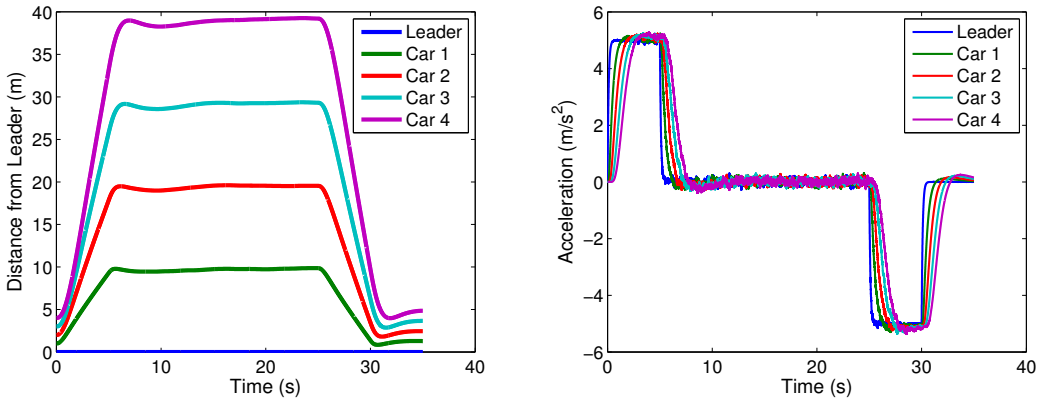


Figure 4.5: In this figure, we show the system operating under noisy conditions with the variance in noise set to .001 of the vehicles velocity. On the left we show the distance from the leader and on the right we show the acceleration for each of the vehicles.

and

$$\phi_6 = \begin{pmatrix} 1 & 0 & 0 & 0 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & 1 & \dots & 0 \\ \vdots & & & & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 0 \end{pmatrix}. \tag{4.10}$$

We used the linear double integrator for modeling the vehicles in the system but in an actual implementation more advanced models can be used. The techniques used for modeling can be as complex as desired considering trade-offs in accuracy, calculation cost, and time for calculation. Improvements that may be considered in the modeling include capturing non-linear behavior of the vehicles drive-train and using terrain mapping to predict variations.

4.3.2 Thresholding Techniques

Once we have a model of $\bar{x}_{i-1}|\hat{u}_{i-j}$ we can then predict whether the model error is acceptable or not. We indicate the measured values for $\hat{x}_{i-1,m}$ and assume we can measure acceleration and velocity. It is not possible to measure the error since we do not have a line of site to car $i-2$.

We propose using the model error normalized to acceleration such that

$$(a_{err}|\hat{u}_{i-j}) = \left(\frac{(a_{m,i-1}|\hat{u}_{i-j}) - a_{i-1,}}{a_{i-1}}\right)^2 \tag{4.11}$$

$$(v_{err}|\hat{u}_{i-j}) = \left(\frac{(v_{m,i-1}|\hat{u}_{i-j}) - v_{i-1}}{a_{i-1}}\right)^2 \tag{4.12}$$

$$(\hat{u}_{err}|\hat{u}_{i-j}) = \left(\frac{(\hat{u}_{m,i-1}|\hat{u}_{i-j}) - \hat{u}_{i-1}}{a_{i-1}}\right)^2 \tag{4.13}$$

We use a hand-tuned constant delay between the model and measured data since the model trails the real system.

4.3.3 Attack Mitigation

We propose a two-state operating condition for the monitoring vehicle. The vehicle uses the CACC controller proposed in Section 3.1.1. When an attack is detected the control law changes to a non-adaptive cruise control law such that $u_i = u_{fb,i} = k_d\dot{e}_i + k_p e_i$ where the error is now calculated with a larger headway constant, for example 1 second. In Section 4.4, we show that this controller is effective at mitigating the impact of the collision induction attack, avoiding the loss of life or assets. This controller would cause other cars in the platoon to flag the detecting car as an attacker and result in the loss of the platoon formation.

As a design decision other reactions could be implemented to mitigate the impact of abnormal behavior. For example a control law can be designed where the headway is proportional to the amount of error in the model and actual system. We leave the design of more response techniques as an implementation decision.

Global reputation system

The ability to detect malicious and abnormal behavior can be combined with a global system to keep a reputation for vehicles. If a car continually gets flagged then it can be investigated by authorities. Similarly, a car that is often flagged can run a diagnostics routine to determine if it

has a system failure as proposed in various works on reputation systems [105].

4.4 Simulations

We simulate our attacks as well as the detection scheme to provide a proof-of-concept in a five car platoon. We use the following parameters for our platoon: $\eta = 0.1$, $h_d = 0.35s$, $k_p = 0.2$, $k_d = 0.7$, and $K = 5$ cars. We set the sampling time for the radar at $T_s = 0.001s$ and assume that the update for the feedforward information occurs every $100ms$. For each trial we assume that the lead car in the platoon accelerates from standstill for 5 seconds at a constant rate, maintains the maximum speed for 20 seconds, decelerates at a constant rate for 5 seconds, and remains at rest for 5 seconds. We do not make assumptions on the acceleration rate used in the test. We assume a model delay of $250ms$, which was determined by empirical tuning.

We assume that the 4^{th} car in the platoon mounts the attack so $a = 3$. We assume that the 5^{th} car is monitoring for the attack and can react if an attack occurs. The monitoring car receives DSRC communications from the 2^{nd} and 3^{rd} advertising their respective acceleration profiles. Thus car 5 has a model

$$err = \begin{pmatrix} a_{err}|\hat{u}_1 \\ v_{err}|\hat{u}_1 \\ \hat{u}_{err}|\hat{u}_1 \\ a_{err}|\hat{u}_2 \\ v_{err}|\hat{u}_2 \\ \hat{u}_{err}|\hat{u}_2 \end{pmatrix} \tag{4.14}$$

to base its detection results on. We hand tune our detection parameters to $\delta = [0.23, 0.48, 0.9, 0.46, 0.9, 0.9]^T$. We assume that if any element of $err > \delta$ the system is under attack and instantly switch to an ACC.

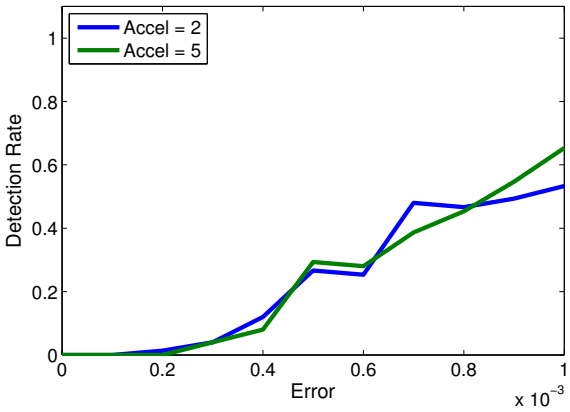


Figure 4.6: In this figure, we show the false positive rate for different levels of noise with velocity relative variance.

4.4.1 False Positives

We consider the impact of noise on our detection scheme by adding Gaussian noise to the acceleration. We assume that the variance in system noise is proportional to current velocity of the car. In Figure 4.5 we show the system during noisy operation with the variance for the car i set to $\sigma_i = .0005v_i[k]$. This results in the acceleration update equation being modified to $a_i[k+1] = a_i[k] + v_i[k] + \mathcal{N}(0, .0005v_i[k])$.

We simulate the system without an attacker present to explore the impact of noise on the false positive rate of our detection scheme. In Figure 4.6 we model two acceleration rates ($2\frac{m}{s^2}$ and $5\frac{m}{s^2}$) at various noise levels. For each noise level and acceleration profile we run 75 trials and calculate the percentage of time that an attack was detected. In this figure, the false positive rate is acceptable when the noise variance is under $.0004$ of the velocity. When the noise increases beyond this the false positive rate is extremely high.

We can mitigate the high false positive rate by using filtering to limit the impact of Gaussian noise.

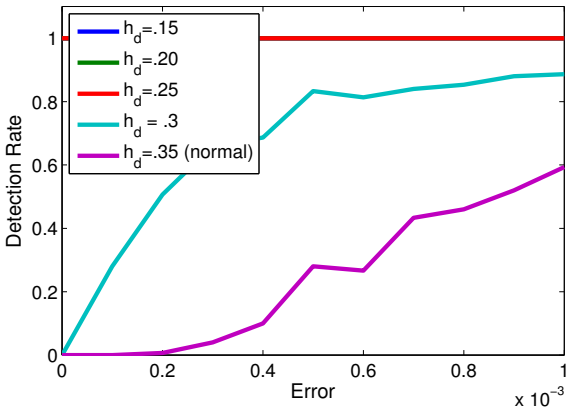


Figure 4.7: In this figure, we show the headway attack detection results, we calculate the false positive rate across 75 trials with an acceleration rate of $2\frac{m}{s^2}$ and 75 trials with an acceleration rate of $5\frac{m}{s^2}$.

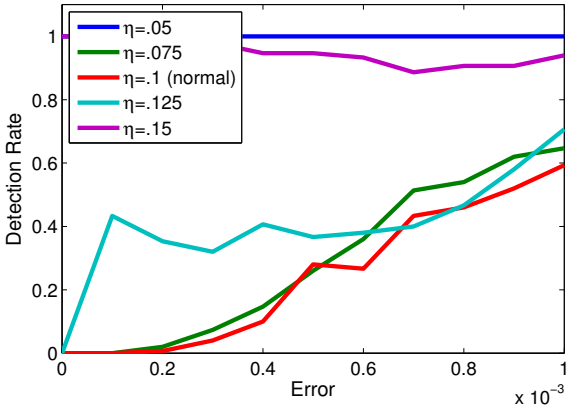


Figure 4.8: In this figure, we show the abnormal behavior detection results, we calculate the detection rate across 75 trials with an acceleration rate of $2\frac{m}{s^2}$ and 75 trials with an acceleration rate of $5\frac{m}{s^2}$.

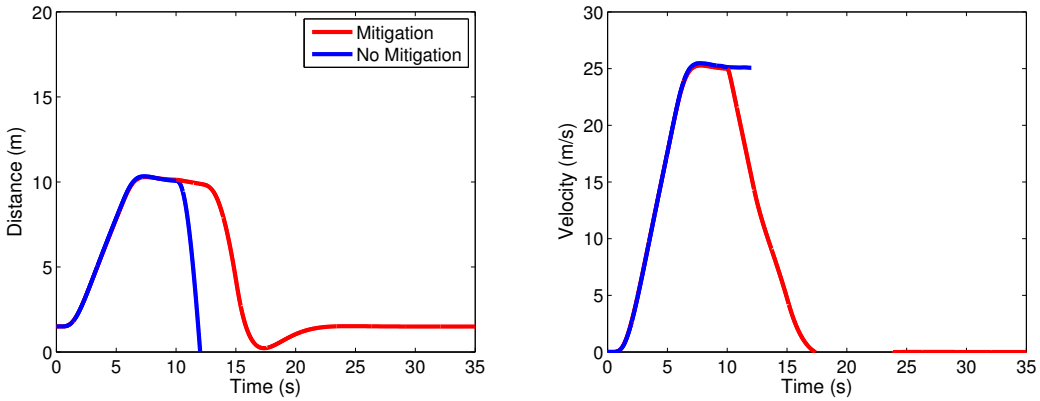


Figure 4.9: In this figure, we show a car using our attack detection technique avoiding an accident during a collision induction technique. We plot distance between the attacker and monitoring on the left and the velocity of the monitoring vehicle on the right. The blue line, which represents the outcome without the detection scheme, stops when a collision occurs. We provide a video simulation of the collision avoidance at `http://goo.gl/o2Uqn7`.

4.4.2 Attack Detection Results

We again simulate the attacks at various noise levels and, where applicable, with various attack parameters. Similarly, with mis-report percentages $\beta \in [0.05, 0.1, 0.15, 0.2, 0.3]$ we detect 100% of mis-report attacks. Additionally, collision induction attacks are detected 100% of the time. In the next section we explore whether they are detected fast enough to mitigate the attack's effect.

In Figure 4.7 we show the result for detecting headway reduction with our detection scheme. For headways under 0.3 seconds the detection scheme works 100% of the time. when the headway is close to the expected value of 0.35 seconds, i.e. at 0.3 seconds the detection scheme is not effective.

In Figure 4.8, we show the results for detecting abnormalities in the lag using our detection scheme. We again calculate the detection rate based on 75 trials with $2\frac{m}{s^2}$ acceleration and 75 trials with $5\frac{m}{s^2}$ acceleration. When $\eta = 0.075$ and $\eta = 0.125$ which is close to the expected value of $\eta = .1$ the attack is not detected. This is desired behavior since the lag parameters are defined by a range in real system. When the values of η are 50% greater or less than the expected value the detection rate is over 90%.

Regardless of noise level our detection algorithm does not detect when a car attempts to join without a radar. This is expected as the vehicle that joins without a radar uses a model based acceleration profile, matching the model-based acceleration that we compare against extremely well.

4.4.3 Attack Avoidance Results

Lastly, we explore whether a collision induction attack can be detected quickly enough to mitigate the collision. In Figure 4.9 we show the performance of a car using our detection scheme when a collision induction attack is mounted at 10.001 seconds with no system noise. We provide a video simulation of the collision avoidance at `http://goo.gl/o2Uqn7`. The blue line shows the behavior when no detection and mitigation scheme is used. It is clear that just after 12 seconds an accident occurs at $25\frac{m}{s}$. The red line shows the same attack when the detection and mitigation scheme is used. It takes less then $100ms$ to detect the attack and as can be seen no collision occurs, meaning the detection coupled with the non-cooperative ACC controller successfully mitigated the attack.

4.5 Discussion

In this chapter, we introduce a technique for a platoon of cars to mitigate the impact of an attacker by detecting and reacting to abnormal behavior. This is an aggressive approach to security, but when dealing with critical cyber-physical systems this may be a reasonable precaution. This technique is also able to discover when non-malicious abnormalities exist, combined with a reputation system this can allow for cars to collaboratively suggest vehicle maintenance needs.

This defense strategy is an example of a defense when no knowledge of the attacker is available but the system is known. This is a likely scenario for many defense application, including those involving cyber-physical systems.

4.5.1 Limitations and Future Work

It is important to choose the detection threshold for the system carefully balancing false negatives and false positives. Given that a false negative can have a catastrophic outcome, either loss of life or loss of personal assets. The cost of false positives on the other hand is relatively low, 20-70 % decreased headway and 5% decreased fuel efficiency, making the costs of a false positive relatively low. Given the danger of false negatives and relatively low cost of false positives, a designer likely chooses a conservative model.

A second important design decision is how to model the vehicles given DSRC packets. In the real system, this involves modeling the dynamics of the car and the interaction with the environment; for example a car performs differently going up hill. The cars have to have a trusted way to choose parameters for each car either through a cloud-based service or through a trusted broadcast scheme. Modeling the dynamics with the environment can easily be supplemented with GPS data to estimate environmental impact.

Another important design decision is how to respond when an attack is detected. In this work, we assume that all attacks are equally bad and when anomalous behavior is detected switch to ACC but many other approaches can be used. For example, a designer can use an adaptive response where the headway and feedforward controller weight are adjusted based on the anomalous behavior measurement.

One major limitation of our approach is the inability of our scheme to detect when the platoon leader is misbehaving. To mitigate this attack we have to consider global schemes with trusted and secured cars that mitigate risk. Our system is also susceptible to noise. To mitigate this limitation we can use many noise reduction techniques. For example, we can mitigate the impact of Gaussian noise in our observations by using a Kalman filter for optimal estimation.

Another limitation of our detection scheme is its inability to detect when a vehicle joins without a radar. This behavior is expected because when a vehicle joins without radar it uses a model based approach to determine its acceleration. Thus the model based approach to the non-radar

equipped vehicle performs similarly to the system model based on future communications. In normal scenarios this performs acceptably but becomes dangerous when the DSRC is unavailable. One approach to detect a car without radar is to occasionally introduce a small amount of noise at a given car's velocity. The car directly behind it responds according to the feedback controller if it has a radar. After a short period the first noise inducing car can communicate how it inserted noise and it can be tested.

It is also noted that our scheme likely causes the platoon to disassemble around the detecting car. This opens up another attack similar to the efficiency attacks [2] allowing a car to lessen the efficiency of the platoon.

This technique is able to detect malicious and abnormal behavior because the system is known. One direction of future work is to investigate what can be done when the attacker is unknown and the system is either stochastic or unknown. Detecting abnormalities in a stochastic system is a logical next step. Detecting abnormalities in an unknown system is another interesting possibilities but the feasibility of such research has to be explored.

The scenario of a known system with unknown opponent can also be explored from the attacker's perspective. In this case, an attacker may desire to destabilize a system, but not know the defensive capabilities. It would be interesting to explore what an attacker can do to perturb the defensive capabilities of a system without getting neutralized.

Chapter 5

Attacks and Defenses Against Rational Opponents with a Known System

In Chapter 3, we discussed how an attacker can leverage full knowledge of a system and the defender's action to design devastating attacks. In Chapter 4, we discussed how a defender can safely operate when no knowledge of the attacker is available. The full knowledge and no knowledge scenarios are reasonable for some cases but many times opponents have to make assumptions about their opponents strategy. In this chapter, we discuss attack and defense strategies of rational players using a 2-player energy-matched jamming model. Using dynamic programming we find a Nash Equilibrium strategy to this game and compare this strategy to adaptive, random, and constant strategies.

As discussed in Section 1.1.2, software-defined radio (SDR) has lowered the barrier on agile wireless communications, destroying common assumptions that jamming is power-agnostic [106] and allows for intelligent power-efficient jamming attacks [19, 20, 21, 22, 23, 24]. Fortunately, the same innovative technologies that enable energy-efficient and stealthy attacks can also enable more robust and agile anti-jamming techniques. The agility provided by SDRs allows defenders and attackers alike to have more fine-grained control of protocols and parameters, enabling the ability to adapt on the fly [25]. However, this mutual agility increases system complexity

and presents a significant challenge to our understanding of various performance, security, and reliability metrics required for effective system design. Understanding how mutually agile opponents interact in a resource-constrained scenario remains an active research field. In this work, we explore a battery-operated jammer and battery-operated sender, where the sender's goal is to successfully transmit and the jammer's goal is to prevent that.

To increase our understanding of rational, mutually agile, resource-constrained players, we look to game theory for tools to analyze optimal strategies for jamming and anti-jamming. We make the following contributions in this chapter.

- We design a new model for energy-constrained jammer-defender interaction, allowing players to transmit or sleep during any round. This provides for the exploration of a realistic scenario where opponents have similar energy levels and freedom to reconfigure.
- We model this interaction as a zero-sum finite-horizon stochastic game with deterministic transitions to find optimal player strategies. We leverage the properties of our game to design a simple polynomial-time dynamic programming algorithm that solves a series of small linear programs to compute optimal strategies (a Nash equilibrium).
- We implement a simulation of three scenarios to gain insights on the performance of energy-constrained Direct Sequence Spread Spectrum (DSSS) and Frequency Hopping Spread Spectrum (FHSS) systems.

Our first contribution is considering attacking and defending opponents that can choose (1) whether to transmit or sleep, (2) what power level to transmit with, and (3) what channel (or how many channels) to transmit on, with the understanding that each choice has a different energy usage and that the outcome also depends on the action chosen by their opponent. Due to the energy constraints and the fact that every choice has a non-zero cost (even sleeping is subject to non-trivial energy leakage), both players can only participate for a finite amount of time. Moreover, since the value of data may significantly decrease with latency, we allow the sender's utility to decay with time. Since our game incorporates aspects of power control and sleeping

for throughput and latency management, we refer to our game formulation as a *finite-energy jamming game*. We are the first to mathematically model and analyze accumulative energy-constrained jammer-sender strategic interaction. In related energy-constrained work, the focus has been on average energy consumption [107] or over-heating [89].

In particular, our finite-energy jamming game formulation imposes maximum energy expenditure on both the jammer and defender, while allowing both players to adjust their transmit power levels. The sender and receiver, collectively comprising the defender in our scenario, communicate either using single-channel DSSS or multi-channel FHSS. We model DSSS and FHSS because many modern communication systems use variants of these techniques. Likewise, the attacker can jam on one or many channels depending on which technique is employed by the defender. If the sender selects a sufficiently high power level compared to the jammer's selected power level (or if the jammer chooses a different channel), then the transmission is successful. Regardless, both player's expend an amount of energy that corresponds to their chosen power level (or sleeping).

In most related work on modeling jamming games, energy constraints have not been considered, so single-shot or repeated game approaches have been adopted. In contrast to that work, the energy constraint means that our game has state, and thus needs more advanced modeling techniques. The two papers in the literature that are closest to our work in this chapter are the following. First, Altman et. al. [107], consider jamming in a stochastic game setting. Whereas we assume that actions and energy levels are fully observed, their work goes the opposite direction and requires that actions are completely unobserved. The truth lies, of course, somewhere in the middle, but both our and their work can shed light on the possibilities and limitations for the general problem. Second, Mallik et. al. [89] consider a dynamic game where temporal energy constraints exist, in the form of over-heating. This means that energy usage only impacts the immediate rounds afterwards, as opposed to expending energy from a finite supply. Like us, they assume that actions are fully observed. They propose a dynamic game, almost akin to a repeated

game, with slight variations in the available actions.

Our second contribution is an algorithm that computes optimal strategies for our system, formulating it as a zero-sum finite-horizon stochastic game with deterministic transitions. We use Nash equilibria as our framework for optimal strategies. Nash equilibria are a compelling solution concept especially for zero-sum settings such as ours, as they guarantee the highest utility against optimal opponents and sub-optimal opponents only increase our utility. As such, Nash equilibria and their associated expected utility represent the best guarantee on utility that one can hope for, when faced with potentially optimal adversaries.

We leverage the zero-sum and finite-horizon properties to design a simple polynomial-time dynamic programming algorithm that solves a series of small linear programs to compute a Nash equilibrium. The dynamic programming aspect is similar to the work of Mallik et. al. [89], who also use the finite-horizon aspect to obtain a dynamic programming description. However, they further use specific properties of their setting to derive analytical solutions, whereas our work relies on algorithms for computing strategies. Their consideration of temporal constraints could easily be incorporated into our more general framework and algorithms, along with our finite-resource energy constraints.

Our third contribution is a series of simulations of finite-energy jamming games, which provide insights into robust communication among reconfigurable yet energy-limited radio systems. To further understand the benefit of our game-theoretic models, we compare the rational player using the finite-energy jamming game model with a random player and an adaptive player, demonstrating several cases where the game-theoretic strategies provided by finite-energy jamming game provides significant gains over other strategies. The game theoretic strategies also provides interesting insights about the tradeoffs of energy-constrained jamming-defender interaction. Of particular interest and matching our intuition, we observe that the jammer's optimal strategy is extremely aggressive when the sender highly values low-latency communication, resulting in an attack strategy using high-power jamming in the beginning. This forces the sender

to transmit with low probability in the beginning of the game, even when highly valuing low latency. In addition to these observations, we evaluate a number of different attack and defense scenarios, and identify a number of interesting trends and tradeoffs in the realm of finite-energy jamming games. In order to mimic a realistic scenario, we set the jammer and defender's initial energies to be within an order of magnitude of each other for our simulations.

We introduce our system model and assumptions in Section 5.1, and we present finite-energy jamming games in Section 5.2. In Section 5.3, we present our simulation and evaluation setup, and we discuss our simulation results in Section 5.4. Lastly, in Section 5.5 we briefly discuss limitations, future directions, and this chapter's impact on this thesis.

5.1 System Model and Assumptions

In this work, we explore a three-node scenario consisting of a sender, a receiver, and a jamming attacker over a time interval $\mathcal{T}$, as illustrated in Figure 1.2. The sender and receiver collectively comprise the defender, able to use single-channel DSSS or N-channel FHSS, while all of the defender's decisions in our scenario are made by the sender. We assume that both the attacker and defender are energy constrained, starting with initial energy $E_{a,0}$ and $E_{d,0}$, respectively, so they are forced to balance between maximum performance and minimum energy expenditure.

We assume that the time interval $\mathcal{T}$ is divided into distinct sub-intervals referred to as *rounds*. In each round, the defender chooses a transmission power p_d from a discrete set of power levels $\mathcal{P}_d \subseteq \{0, 1, \ldots, p_{d,\max}\}$, where $p_{d,\max}$ is the defender's maximum transmission power. When the defender transmits with power p_d in a round, it incurs an energy cost $\epsilon_d(p_d)$, and we assume two fundamental properties of this cost function: monotonicity and strict positivity. Monotonicity of the cost function simply means that higher transmission power incurs higher energy cost, while strict positivity means that all actions incur an energy cost, even a play of $p_d = 0$, in which case the defender pays a leakage cost while sleeping. In the case of FHSS, the defender also chooses which of the N channels it will use for communication. We assume that the underlying

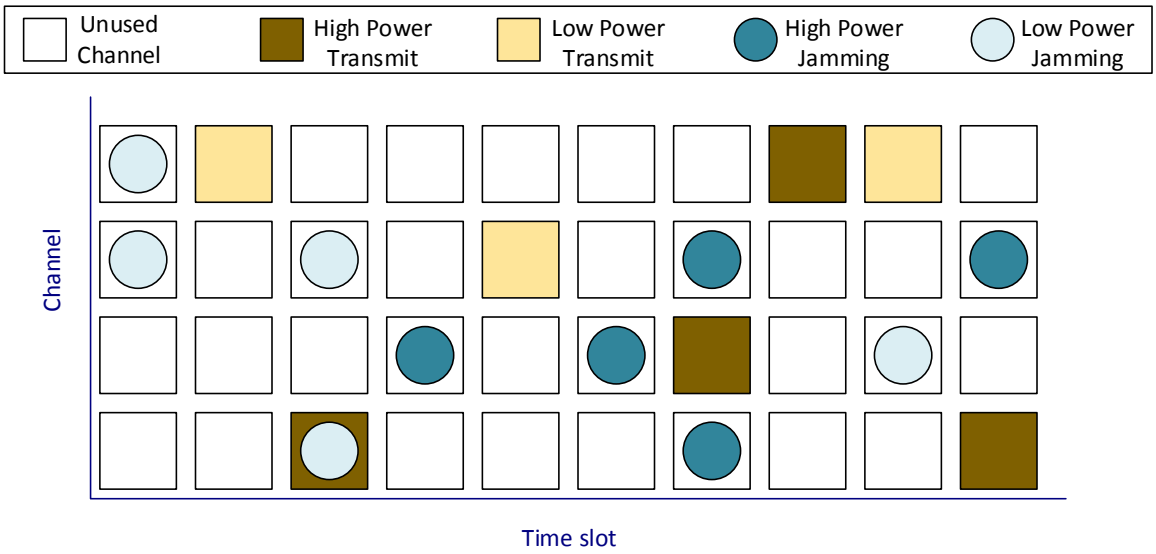


Figure 5.1: We illustrate our system and show the Finite-Energy Jamming game. The jammer and sender both are able to choose to power nap, transmit at a low power, or transmit at a high power.

synchronization, configuration, and channel switching costs are negligible, and thus we treat them as free. We illustrate this system in Figure 5.1.

The attacker's energy model is similar. In each round, the attacker chooses a jamming power p_a from a discrete set of power levels $\mathcal{P}_a \subseteq \{0, 1, \ldots, p_{a,\max}\}$, where $p_{a,\max}$ is the attacker's maximum jamming power. The energy cost of the attacker's action in the round is also dictated by a function $\epsilon_a(p_a)$, which is both monotonic and strictly positive as with the defender, but with one significant difference. In the FHSS case, the attacker is allowed to reconfigure the radio front-end to jam any k out of the N channels, where $1 \leq k \leq N$, using power p_a per channel. The cost for the attacker for this round is then given by $k\epsilon_a(p_a)$ proportional to the number of channels jammed.

To measure the defender's performance in each round, we consider the throughput $T(p_d, p_a)$ achieved in the round when the defender transmits with power p_d and the attacker transmits on the same channel with power p_a. Moreover, since the value of the sender's data to the receiver

may decrease with time, we introduce a discount factor $\delta \in (0, 1]$ for every round in which the data does not reach the receiver. This discount function is representative of a system where the sender has all data at the beginning of the time interval and desires rapid transmission. To compensate for the latency induced by the jamming attack in the defender's utility function, we multiply the throughput $T(p_d, p_a)$ by δ for each round of delay, so any throughput attained during round i is valued according to the *latency-adjusted throughput* $\delta^i T(p_d, p_a)$. When the attacker jams on a channel different from that used by the defender, we use the equivalent throughput $T(p_d, 0)$, since the attack has no effect.

We assume a perfect knowledge scenario starting at the beginning of the time interval, so the players know the initial energy of both players. In addition, we assume that each player can observe their opponent's actions in that round by the end of the round, so each player always has complete knowledge of their opponent's residual energy at the beginning of the next round when they have to decide what to do in that round.

5.2 Finite-Energy Jamming Games

We model a finite-energy jamming game in the described wireless system as a zero-sum finite-horizon stochastic game with deterministic transitions. In much of the related literature, jamming scenarios have been modeled as single-shot or repeated games [108, 109]. Both of these approaches are sensible when no state is present, for example, when energy constraints do not apply, as the same strategy remains optimal throughout the game. Since our game has state in the form of residual energy, neither single-shot nor repeated game models can adequately capture our setting. Instead, we turn to stochastic games, where the game state transitions at every time step. Since residual energy is monotonically decreasing, we can model our game as a finite-horizon game. We allow all transitions between states to be deterministic, since the energy cost of different actions is assumed to be fixed and known.

In two-player zero-sum games, the solution concept of Nash equilibria is particularly com-

pelling. In general-sum games, there can be many Nash equilibria with different expected utilities for the players, and playing a Nash equilibrium strategy says nothing about the expected utility for a player, if the opponent does not play a best response. This is not so for zero-sum games, where playing a Nash equilibrium strategy guarantees at least a certain level of utility in expectation. That guaranteed utility is called the value of the game, and it is achieved when the opponent responds optimally to the player's optimal (Nash equilibrium) strategy. The zero-sum property guarantees that the player can only benefit (and get more than the value of the game) if the opponent does not play optimally. We will formally define a Nash equilibrium in Section 5.2.1.

We first introduce our game framework in the context of the single-channel communication system. We then extend our study to include FHSS with the jammer transmitting on a fixed number of channels. After this, we explore a further extension using FHSS where the attacker can vary the number of jammed channels at each time step. For the FHSS settings, we are not assuming that a single channel is chosen at each time step. Rather, we assume that the frequency hopping is so effective that the best the jammer can do is jam a random subset of channels in the hopes of disrupting communication. Finally, we show how to compute Nash equilibria for these games.

5.2.1 Single-Channel Game

The first game we explore is the single-channel finite-energy jamming game. This game uses a single DSSS channel and has the attacker and defender select power levels from a discrete set. The parameters to the game are the discount factor δ and the initial energies $E_{d,0}, E_{a,0}$ for the players. Based on the defender's residual energy E_d at the start of a round, the defender's action set $A_d(E_d)$ for that round is defined as $A_d(E_d) = \{p \in \mathcal{P}_d : \epsilon_d(p) \leq E_d\}$. The attacker's action set $A_a(E_a)$ is similarly defined. When the defender and attacker choose respective actions $p_d \in A_d(E_d)$ and $p_a \in A_a(E_a)$, the immediate utility to the defender is $u_d(p_d, p_a) = T(p_d, p_a)$, which is later discounted by δ^i during round i to compensate for latency. The attacker's imme-

diate utility is $u_a(p_d, p_a) = -u_d(p_d, p_a)$. The defender chooses its action based on an energy-dependent *strategy* $\sigma_d^{E_d,E_a}$ that specifies a probability distribution over actions in $A_d(E_d)$. For example, $\sigma_d^{E_d,E_a}(p)$ is the probability the defender will transmit at power level $p \in \mathcal{P}_d$. We analogously define the attacker's strategy $\sigma_a^{E_d,E_a}$. Once the players choose their actions in a round i, with the defender and attacker respectively transmitting at power levels p_d and p_a, the game transitions to round $i+1$, where the players have residual energy $E_d - \epsilon_d(p_d)$ and $E_a - \epsilon_a(p_a)$. The game continues in this way until the defender's residual energy is such that $A_d(E_d) \subseteq \{0\}$, after which $u_d = u_a = 0$.

Considering the entire game over multiple rounds, a *strategy profile* σ is a pair of strategies $\sigma = \{\sigma_d, \sigma_a\}$ that fully specifies the game. Using the strategy profile σ, we can then compute the defender's total expected utility $u^\sigma(E_{d,0}, E_{a,0})$ using a recursive definition over diminishing energy levels as

$$u^\sigma(E_d, E_a) = \sum_{p_d \in A_d(E_d)} \sum_{p_a \in A_a(E_a)} \sigma_d^{E_d,E_a}(p_d)\sigma_a^{E_d,E_a}(p_a) \times \Big(u_d(p_d, p_a) + \delta u^\sigma(E_d - \epsilon_d(p_d), E_a - \epsilon_a(p_a)) \Big) \tag{5.1}$$

where $u_d(p_d, p_a) = T(p_d, p_a)$ for the single-channel game. This can be viewed as a series of normal-form games, where the payoff matrix for each game depends on the values of the subgames induced by the various choices of actions.

A Nash equilibrium is a strategy profile $\sigma^* = \{\sigma_d^*, \sigma_a^*\}$ that satisfies

$$\sigma_d^* = \arg\max_{\sigma_d} u^{\{\sigma_d, \sigma_a^*\}}(E_{d,0}, E_{a,0})$$

$$\sigma_a^* = \arg\max_{\sigma_a} u^{\{\sigma_d^*, \sigma_a\}}(E_{d,0}, E_{a,0})$$

In other words, in a Nash equilibrium, each player maximizes their own utility, given the strategy of the other player.

5.2.2 Multi-Channel Game with FHSS

We next consider a finite-energy jamming game in which the defender spreads its transmissions over N orthogonal channels by choosing a different channel randomly in each round of the game. In our first FHSS-based game, the attacker chooses k channels to jam every round, where k is constant for the duration of the game. In each round, the attacker has a probability of k/N of interfering with the defender's transmission, so the immediate utility for the defender in this case is given by

$$u_d(p_d, p_a) = \frac{k}{N} T(p_d, p_a) + \frac{N-k}{N} T(p_d, 0) \tag{5.2}$$

The defender's total expected utility is given by substituting (5.2) into (5.1). In this game, the energy expenditure of the attacker is increased by a factor of k, meaning that an attack action with power p_a incurs a cost $k\epsilon_a(p_a)$.

5.2.3 Multi-Channel Game with FHSS and Selection of Number of Channels to Jam

Similar to our second game, we consider a generalization of the previous FHSS game in an N-channel communication system. In our second FHSS-based game, the attacker is free to choose any value of $k \in \{1, \ldots, N\}$ in each round as part of its attack strategy. Given the additional game parameter, the attacker's action set $A_a(E_a)$ in each round is extended to

$$A_a(E_a) = \{(p, k) \in \mathcal{P}_a \times \{1, \ldots, N\} : k\epsilon_a(p) \leq E_a\}$$

and the utility function $u_d(p_d, p_a)$ is extended to $u_d(p_d, p_a, k)$, using the same form as (5.2). In contrast to the previous game with fixed k, treating k as a variable game parameter allows the attacker to effectively balance the tradeoff between higher utility and greater energy expenditure of jamming more channels. In addition, since the attacker's action set $A_a(E_a)$ has increased

in dimensionality compared to the fixed-k case, the complexity of solving the game increases linearly in N.

5.2.4 Computing a Nash Equilibrium

For each of the three game models described above, we can use the same basic approach for computing a Nash equilibrium. Each of those three games can be viewed as a series of normal-form games, each of which depend on the values of subgames to fill out their payoff matrix. We use this subgame property, along with the well-known fact that zero-sum normal-form games can be solved in polynomial time using linear programming, to solve our problem. Using dynamic programming, solutions are constructed bottom up through successively solving linear programs that compute Nash equilibria of subgames. The pseudocode is presented as Algorithm 1.

Input: Energy levels E_d, E_a, discount factor δ
Output: Nash equilibrium strategy profile σ
$U \leftarrow [\,]$ // dynamic programming table
for $E'_d \in \{0, \ldots, E_d\}$ **do**
  for $E'_a \in \{0, \ldots, E_a\}$ **do**
    $M \leftarrow [\,]$ // payoff matrix
    for $p_d \in A_d(E'_d), p_a \in A_a(E'_a)$ **do**
      $M[p_d, p_a] = u(p_d, p_a) + \delta \cdot U[E'_d - \epsilon_d(p_d), E'_a - \epsilon_a(p_a)]$
    end
    $U[E'_d, E'_a] = \textsc{GameValue}(M)$
    $\sigma^{E'_d, E'_a} = \textsc{StrategyProfile}(M)$
  end
end

Algorithm 1: Bottom-up dynamic program for computing Nash equilibria in finite-energy jamming games.

The dynamic program iterates over all possible energy levels for the two players, starting from the smallest levels possible. For each pair of energy levels, a payoff matrix M is computed. Line 1 implements the recursive equation for utility given in (5.1) or (5.2) depending on the game played. That is, it sets the payoff to the immediate payoff achieved from the actions taken plus the value of the subgame reached by the power loss, weighted by the discount factor δ. Lines 1

and 1 extract the value of the game and a strategy profile that achieves a Nash equilibrium.

Our dynamic program crucially relies on the fact that every set of energies E_d, E_a induces a subgame, where the path traveled to get to these energy levels does not matter. Depending on the round where the energy levels are reached, the discount factor might be different. However, in terms of computing a strategy for E_d, E_a, we can assume without loss of generality that we are at round 0, since for any other round i, every entry in M will be scaled by the same discount factor δ^i, and so the optimal strategies will be the same.

For the function calls GAMEVALUE and STRATEGYPROFILE in Algorithm 1, a solver for computing a Nash equilibrium of M is needed. Since M is a standard payoff matrix for a normal-form game (entries are constants, because values for the subgames have already been computed), we can adopt the standard linear programming approach for computing a Nash equilibrium strategy. We will show how to compute a Nash equilibrium strategy for the defender, with the case for the attacker being completely analogous.

The linear program is shown in Figure 5.2. The variable v denotes the utility for the defender, which is to be maximized. The first two constraints ensure that the defender's strategy at the subgame forms a probability distribution. The last constraint ensures that no matter which action the attacker selects, the defender is guaranteed value v. For any optimal solution, the value of v will be the value of the game, and the computed strategy will be a Nash equilibrium strategy.

$$\max v \tag{5.3}$$

$$\sum_{p_d \in A_d(E_d)} \sigma_d^{E_d,E_a}(p_d) = 1 \tag{5.4}$$

$$\sigma_d^{E_d,E_a}(p_d) \geq 0 \qquad \forall p_d \in A_d(E_d) \tag{5.5}$$

$$\sum_{p_d \in A_d(E_d)} \sigma_d^{E_d,E_a}(p_d) \cdot M[p_d, p_a] \geq v \quad \forall p_a \in A_a(E_a) \tag{5.6}$$

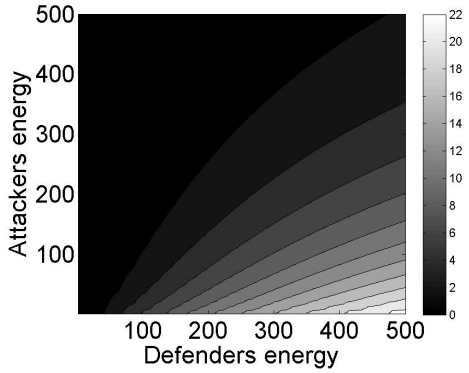
Figure 5.2: The linear program used in computing a Nash equilibrium strategy for the defender. $A_d(E_d)$ and $A_a(E_a)$ are the sets of actions available for the defender and attacker respectively, given their current energy levels.

The number of linear programs can be upper-bounded by the number of possible energy levels in subgames. Given initial energy levels $E_{d,0}$ and $E_{a,0}$, the number of linear programs solved is $O(\frac{E_{d,0}}{\epsilon_d(0)} \cdot \frac{E_{a,0}}{\epsilon_a(0)})$, since the power cost of sleeping divides all other power costs. Each linear program has size $O(|\mathcal{P}_d| \cdot |\mathcal{P}_a|)$.

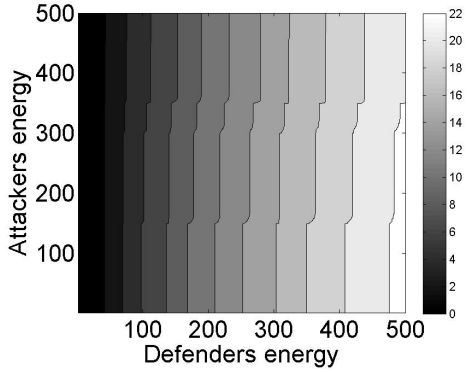
Technically, our algorithm computes a subgame perfect equilibrium, a refinement of Nash equilibria. A subgame perfect equilibrium is a Nash equilibrium such that for any subgame, even those reached with probability zero, the players are playing Nash equilibrium strategies for the subgame. This provides an extra level of robustness over Nash equilibria, as we are not only guaranteed the value of the game, but also guaranteed to play optimally if the opponent chooses a sub-optimal action, assuming the game is played optimally onwards from there. This is not the same as optimally responding to any strategy of the opponent. Rather, it means that we optimally respond to any current game state, assuming that the opponent will play optimally from then on, even with mistakes in the past.

5.3 Simulation

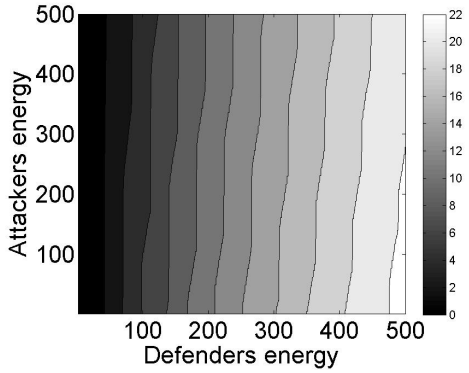
To show the benefits of the finite-energy jamming game we simulate three different games. To use realistic parameters in the simulation we base our parameters on measurement data taken from communication nodes and a jammer implemented with GNUradio on USRP2 software-defined radio. We consider an attacker that is able to adapt their power level and also the number of channels they jam on. The defender is able to choose a channel to transmit on and also choose a power level to transmit at. For our measurements the sender and jammer are connected to the receiver via wire with equal attenuation. This mimics the location of the senders being equidistant from the receiver. In this section, we discuss the parameters we use for our simulation, the optimization results, and the game play simulation we use.



(a) Single channel



(b) Fixed FHSS



(c) Optimal FHSS

Figure 5.3: To demonstrate the optimization, we show the expected utility for 3 different games with varying initial energy levels. For all the games a discount factor of .975 is used and in the frequency hopping game the defender uses 50 channels. The color scale shows the utility of the game.

5.3.1 Game parameters

We take RF power measurements at the connection port and find that the power expended for a low-power attack as $1.16\mu w$ and for a high-power attack as $3.22\mu w$. We also assume a continuous energy drain per round that we estimate as $.5\mu w$. This constant drain controls for calculation, battery leakage, and other constant sources of drainage. We normalize the cost of energy usage per round and define $\{1, 3, 7\} \in \mathcal{P}_a$ as the values for sleeping, low power, and high power attacking, respectively. The jammer is able to simultaneously jam on multiple channels during any round. We assume a linear cost increase per channel for the low- or high-power attacks. Sleeping does not use channels so we assume it has no increased costs.

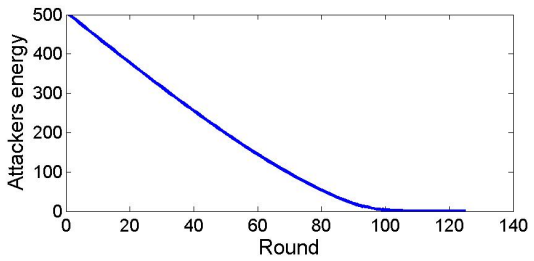
Likewise for the defender we find power at the port for a low-power transmission as $6.5\mu w$ and a high-power transmission as $7.83\mu w$. We again assume a constant energy drain of $.5\mu w$. Normalizing and approximating the cost per round of each play we find costs of $\{1, 14, 16\} \in \mathcal{P}_d$ for sleeping, low power transmissions, and high power transmissions, respectively. We assume that synchronization and key-sharing is done beforehand and that there is no extra cost for the sender to use frequency hopping.

If the defender is transmitting we assume a constant rate so the normalized throughput per round is approximated by packet delivery ratio (PDR). Because of this we use packet delivery ratio in lieu of throughput when the defender is transmitting and assume zero throughput when the defender is sleeping. We measure packet delivery ratio in our single channel 802.15.4 system as

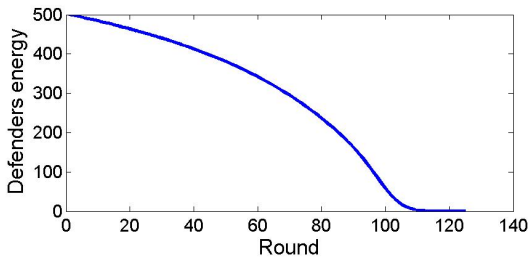
$$\mathrm{pdr}(\mathcal{P}_a, \mathcal{P}_d) = \begin{pmatrix} 0 & .96 & 1 \\ 0 & .58 & .92 \\ 0 & 0 & 0 \end{pmatrix} \tag{5.7}$$

where the attacker is the row player and the defender is the column player. We assume that there is no cross-channel interference so if the jammer is not attacking a particular channel there is no added interference.

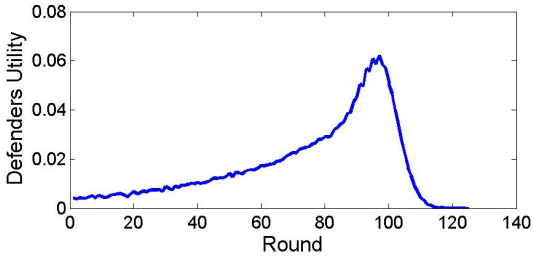
In order to mimic both players using the same class of devices, we constrain the attacker and



(a) Attacker's energy



(b) Defender's energy



(c) Defender's utility

Figure 5.4: The average over 10,000 runs of a simulation of the single channel game with two rational players and a .975 discount factor.

defender to have similar initial energy resources. We define similar initial energy resources as both players having an initial energy that is within one order of magnitude of the other.

5.3.2 Optimization

We use Algorithm 1 to arrive at an optimal strategy and expected utility. In Figure 5.3(a) we show the values for the single channel game with a .975 discount factor. In Figure 5.3(b) we show the expected defender utility for a defender with 50 channels and an attacker with 50 channels and .975 discount. In Figure 5.3(c) we show the defender's utility when the defender has 50 channels and the attacker optimizes power and number of channels. The optimization provides confirmation of what is intuitively expected. The single channel game heavily favors the attacker while either of the frequency hopping games with 50 channels favors the defender.

5.3.3 Game play

We designed a simulator to explore the performance of our computed strategies and compare them to other strategies. Other strategies we use for comparison include a constant strategy, a uniform random strategy, and a weighted average algorithm [110]. The random strategy that we consider uniformly samples from all possible strategies. The weighted average algorithm was designed for a similar power game. It works by keeping a weighted vector of the likelihood of their opponents strategy as well as a matrix of the expected utility for given combinations of plays. The player then uses these to compute their strategy.

We designed a simulator for each of the games introduced in Section 5.2. For the single channel, input parameters include both players' strategies as well as the initial energy of both players, and the discount factor for the players. In the frequency hopping spread spectrum case with a constant number of attacker channels the simulator also takes the number of channels used by the defender N and attacker k. The simulator also accepts the precomputed optimal strategy for both players for the given game and the discount factor.

To demonstrate the operation of our simulator we show the average run of 10,000 trials of the single channel game with a .975 discount factor and two rational players in Figure 5.4. The initial energy for assigned to both player is 500 units in this experiment, and the power levels and corresponding energy usage are given in Section 5.3.1. Figures 5.4(a) and 5.4(b) show the average remaining energy for the attacker and defender, respectively, at the given time. Figure 5.4(c) on the other hand shows the average instantaneous utility for the defender.

To simulate the frequency hopping game the defender selects one channel $n \in [1, N]$ at the beginning of every round. Similarly, the attacker selects k of N channels to interfere with. If n is one of the k channels selected then the attacker is successful and the throughput is calculated

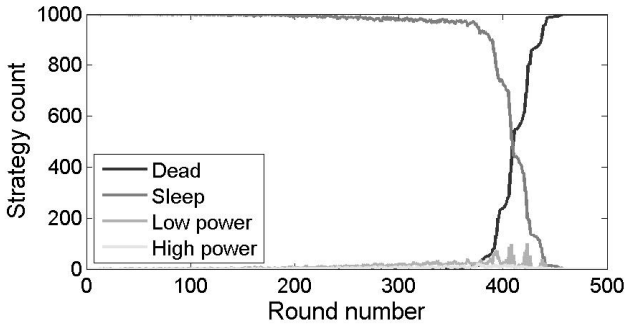


Figure 5.5: Counts of how many times out of a thousand a defender chooses a strategy against a constantly sleeping attacker with a .9 discount factor. We define a dead node as a node that has expended all of its energy.

using (5.7). Otherwise the throughput is calculated using

$$\mathrm{pdr}(\mathcal{P}_a, \mathcal{P}_d) = \begin{pmatrix} 0 & .96 & 1 \\ 0 & .96 & 1 \\ 0 & .96 & 1 \end{pmatrix} \tag{5.8}$$

5.4 Simulation Results

In this section, we discuss simulated scenarios using the setup and parameters presented in Section 5.3. We explore all three games from Section 5.2 and describe insights gained from various experiments.

5.4.1 Single-Channel Game

For the single channel game we compare the performance of the rational, random, and adaptive weighted average algorithm for both players. In Table 5.1 the utility averaged over 100,000 runs for various attacker and defender strategy pairs is presented. Both players start with 200 units of energy and choose optimal strategies for the given discount factor. The results for the rational player always outperform the random and weighted average player's performance, but sometimes this is less pronounced. Although the gain from rationality is marginal with the .9

		Defense								
		.9 Discount Factor			.95 Discount Factor			.99 Discount Factor		
		Rat.	Rand.	Weig.	Rat.	Rand.	Weig.	Rat.	Rand.	Weig.
Attack	Rational	0.258	0.257	0.256	1.242	1.235	1.237	6.024	6.003	6.012
	Random	0.258	3.296	3.143	1.245	4.721	4.539	6.080	7.039	6.887
	Weighted	0.258	2.597	2.529	1.250	3.545	3.444	6.425	5.025	4.957

Table 5.1: Mean defender's utility for the single channel game.

		Defense								
		.9 Discount Factor			.95 Discount Factor			.99 Discount Factor		
		Rat.	Rand.	Weig.	Rat.	Rand.	Weig.	Rat.	Rand.	Weig.
Attack	Rational	0.096	0.312	0.294	0.368	0.769	0.726	1.019	1.373	1.359
	Random	0.265	0.933	0.988	0.510	1.113	1.184	1.098	1.490	1.533
	Weighted	0.255	1.374	1.408	0.505	1.993	2.015	1.184	3.157	3.158

Table 5.2: Standard deviation of the defender's utility for the single channel game.

discount factor, a second factor to consider is that rationality decreases deviation of results. In Table 5.2 we show that either player playing rationally greatly decreases the standard deviation in utility. This decrease in variance can be a significant benefit for designing secure systems in that it is able to provide performance guarantees and less uncertainty.

The single channel game also provides an interesting insight on the effect of rationality on the defender's utility. In Table 5.3 we see that rational play increases the defender's overall throughput. The smaller the discount factor, the greater the gain in defender's throughput from rationality against a rational attacker.

Another interesting result is highlighted in Figure 5.5. In this figure, the attacker always chooses to power nap while the defender is rational. This results in an attacker that has a slow

		Defense								
		.9 Discount Factor			.95 Discount Factor			.99 Discount Factor		
		Rat.	Rand.	Weig.	Rat.	Rand.	Weig.	Rat.	Rand.	Weig.
Attack	Rational	7.016	0.768	1.019	6.498	1.953	2.154	6.633	6.113	6.124
	Random	6.849	7.207	7.052	6.622	7.208	7.066	6.679	7.200	7.054
	Weighted	6.270	5.147	5.063	6.452	5.150	5.032	7.062	5.1269	5.061

Table 5.3: Defender's mean throughput for the single channel game.

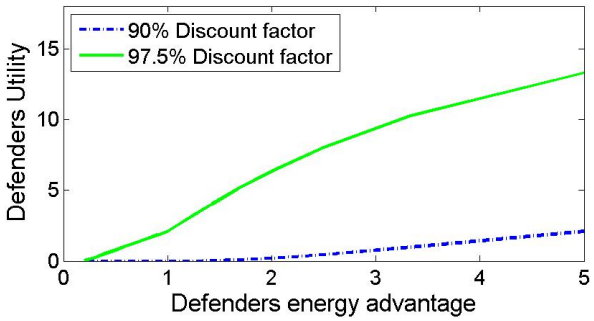


Figure 5.6: Advantage gained by an attacker or defender having a energy advantage with varying discount factors. The advantage shown is the multiplicative advantage such that defender's advantage $= \frac{E_d}{E_a}$.

but constant energy fade. The rational defender plays as if the attacker was also rational, and therefore he transmits with very low probability in the beginning of the game. This is highly counterintuitive from a throughput perspective, since the sender could transmit freely, and gain much higher utility. This is an example of how inoptimal opponents are not exploited optimally by a Nash equilibrium strategy, since the sender has to assume that the jammer might start playing optimally at each round, in order to guarantee attaining the value of the game.

We also explored the effect of a difference in energy between the two players. In Figure 5.6 we show the defender's utility for various advantages in the attacker's energy. The curve here, while qualitatively intuitive, can be instructive in how much extra energy a defender must have to perform well in the presence of an attacker.

5.4.2 Multi-Channel Game with FHSS

The second set of experiments we conduct considers a defender frequency hopping over a set of N channels and a jammer blocking a set of K channels per round. In Figure 5.7 we show the defender's utility for various sets of attacker and defenders channel numbers with a .975 discount factor when both players are rational. This leads to two conclusions when the defender and attacker have similar initial energy. First, a defender with 20 or more channels effectively mitigates the jamming threat. Second, an attacker jamming fewer channels in this case can be

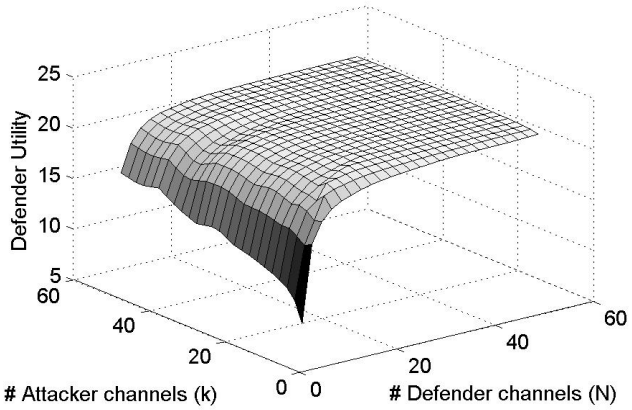


Figure 5.7: Defender's utility for the set number of attacker channel FHSS game. The attacker and defender both choose their power levels optimally for the number of channels they are using.

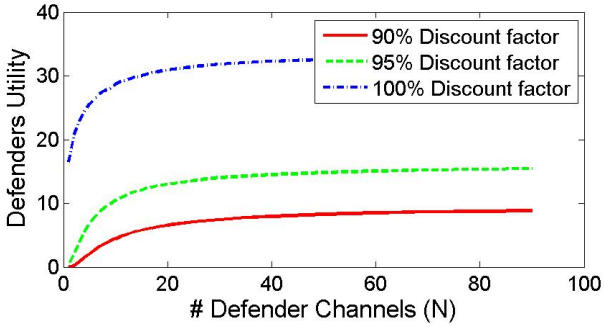
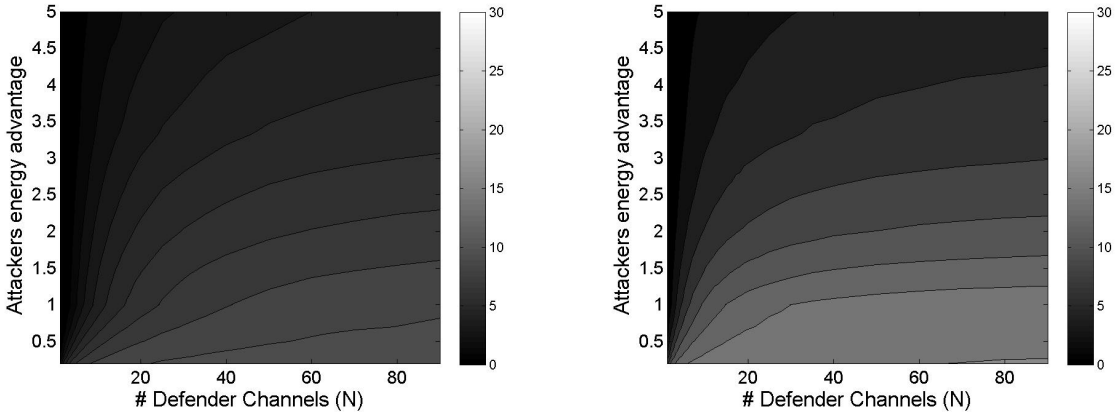


Figure 5.8: In this figure, we show the mean defender's utility for varying numbers of defending channels and an optimal attacker.

beneficial to the attacker. One likely explanation for this is sensitivity to power cost, since only being able to jam a large number of channels (as opposed to being able to vary this) expends a large portion of the energy.

5.4.3 Multi-Channel Game with FHSS and Selection of Number of Channels to Jam

Our third set of experiments considers the FHSS game where the attacker can choose power levels and the number of simultaneous channels to attack. In Figure 5.8 we show the mean defender's utility for various discount factors. This figure suggests that above a certain number



(a) .9 discount factor

(b) .95 discount factor

(c) .99 discount factor

Figure 5.9: Defender's utility for various channel numbers against an optimal attacker. We define the attacker's energy advantage as $\frac{E_a}{E_d}$.

of channels, even with an optimal attacker, there is a diminishing return on investment for the defender adding more channels. The attacker's strategy with the lower discount factor causes an attacker to select a very aggressive strategy, often expending all its energy as quickly as possible in hopes of causing some degradation to the transmission.

In Figure 5.9 we show the defender's utility for various attacker multiplicative energy advantages defined as $\frac{E_a}{E_d}$. These curves allow for a decision of how much of an energy advantage an attacker needs to overcome spread spectrum. This also illustrates that the number of channels a defender needs to be protected from a jamming attack varies on the difference in the two players energy.

5.4.4 Summary of Simulation Results

In this work, we consider three different finite-energy jamming games. The first is a single channel DSSS jamming game, the second is a FHSS game where the attacker jams a constant number of channels, and the third is a power nap game with jammer attacker the optimal number of channels.

In the first game we find that either player playing rational decreases the variance in the game, a beneficial result for designing a secure communication system. We also noted that in this game a rational defender greatly increased the overall throughput of the system. We also showed that rationality can be detrimental to the defender. When the discount factor is small and the attacker chooses a strategy of constantly sleeping the defender is intimidated into not transmitting until most of the energy is drained.

In the second game we show that the a defender that hops over at least 20 channels is effectively able to mitigate the effects of jamming. We also show, counter to intuition, in some scenarios when the attacker jams less simultaneous channels it has a greater impact.

In the third game we confirm the intuition of a diminishing return for the defender past a certain number of channels. We also show the tradeoff in the advantage in the attacker's energy

level and the number of channels. These charts provide a basis for choosing the number of channels a defender needs for hopping based on how much extra energy is available for the jammer.

5.5 Discussion

In this chapter, we introduce a game theoretic model that demonstrates the analysis of rational opponents with a known system. We introduced *finite-energy jamming games*, a game-theoretic framework to understand energy-constrained jammer-defender interaction. We developed several game models within this framework, where the sender and jammer can vary their power levels and whether to send at all. In our more advanced models, we introduced frequency hopping to the game model, and investigated the effect of allowing the jammer to vary number of channels jammed on. To do this, we modeled our system as a zero-sum finite-horizon stochastic games with deterministic transitions. Leveraging the properties of our game, we designed a simple and fast polynomial-time dynamic programming algorithm for computing a Nash equilibrium.

We implemented a simulator to explore the practical properties of our framework across our different game types. Using our simulator, we investigated the possible guarantees that can be achieved under various game settings. We also investigated the practical performance of Nash equilibrium strategies against simpler strategies, such as adaptive or fixed randomized strategies. An interesting result from this analysis was the decrease in variance provided from a rational player, a beneficial property for designing secure systems. Another interesting result provided by this analysis is that an optimal opponent that sleeps constantly still leads to a rational sender incurring large performance losses, due to the assumption that the attacker will play optimally.

5.5.1 Future Work

There are several interesting future research directions to extend this chapter. First, to make the problem more practical, it would be interesting to relax the perfect knowledge assumption and replace it with an observation based approach. This would make the game model significantly harder to solve, and so more advanced computational approached would be needed. Second, expanding this chapter to the setting of multiple jammers and multiple defenders would provide a better understanding of interactions of adversaries in the wild. Third, the scope of both players could be expanded to include multiple layers in the communication stack and cross-layer attacks. Fourth, we could expand this game to realistic models of communication channels that consider their stochastic properties. Fifth, we could consider the use of similar game theoretic analysis when an attacker can choose from a wide range of attacks.

Chapter 6

Attacks Against a Constant Opponent with a Partially Known System

In Chapter 3, Chapter 4, and Chapter 5 we design strategies for a player that knows the system that they are interacting with. In reality, many attackers do not know the exact parameters of the system they are attacking. Not knowing the appropriate parameters can cause an attacker to use excessive resources. A jamming attack using a generator and high-power wide-band jamming to effectively interfere with communications but also is easy to detect and could result in retaliation.

In this chapter, we design an observe-and-adapt approach to estimate the parameters of the system-under-attack which can be used to mount an efficient attack. We make two assumptions in this chapter. First, the defender under attack is constant in their strategy. Second, the system parameters are constant or slow to change. To demonstrate our observe and adapt approach, we design a wireless communication attack known as jamming as discussed in Section 1.1.2 and Section 2.3, and Chapter 5.

Low-power jamming attacks have been designed recently but generally use static strategies against a particular protocol [79, 81] such as short form periodic jamming (SFPJ) [19]. The SFPJ attack uses very short loud bursts to interfere with DSSS communications. Such an attacker is very effective against IEEE 802.15.4, which is commonly used in sensor networking, but requires

tuning against a system and node geometry to obtain good results. In this work, we present Self-Tuning, Inference-based, Real-time jamming or *STIR-jamming* which explores an attacker which continually adapts its attack parameters with performance informations obtained from the system under attack. To do this we envision an attacker with both inference and jamming capabilities. The attacker's inference capability estimates the performance of the legitimate network in real time. Using the inferred information, the attacker then tunes its jamming attack to achieve better performance. This attack differs greatly from traditional jamming in that it looks to continually optimize the physical layer jamming attack in real time. This type of attack can use commodity sensor network hardware to make an efficient and long lasting attack, showing the need for more research in detecting and mitigating low-power jamming attacks against DSSS.

The major contributions of this chapter are as follows.

- We propose the STIR-jamming framework for continually modified jamming attacks using an observe-and-attack feedback loop between the attacker and the target system.
- We present two instances of STIR-jamming. The first algorithm performs repeated parameter optimization using a reference model to predict the effect of parameter selection. The second algorithm iteratively tunes the parameters to increase or decrease the attack impact.
- We show results from a proof-of-concept implementation of the two STIR-jamming algorithms against a link using the 802.15.4 protocol and empirically show these attacks achieve relatively stable performance.

The remainder of this chapter is organized as follower. In Section 6.1, we introduce our system model and in Section 6.2 we introduce the STIR-jamming framework and algorithms. We show how the two STIR-jamming algorithms can be implemented against an 802.15.4 link in Section 6.3 and present empirical results in Section 6.4. Finally, Section 6.5 discusses future work and limitations. We provide a discussion of related work in jamming and anti-jamming in Section 2.3.

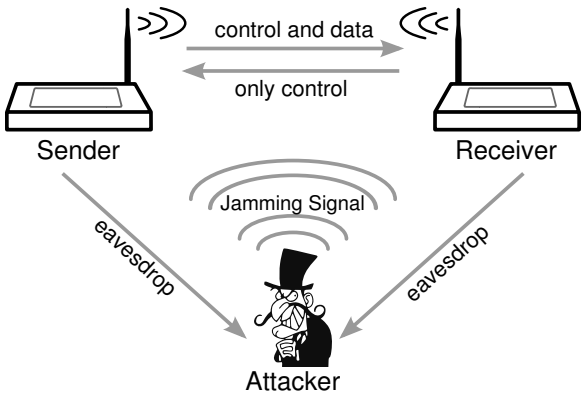


Figure 6.1: Our attack model gives the attacker both observation and jamming capabilities allowing for continual modification of attack parameters from observed performance characteristics.

6.1 System Model

In this section, we introduce the system model and notation used for the remainder of this work. We consider a communication system containing a sender, a receiver, and an attacker as shown in Figure 6.1. The sender transmits both data and control packets, and the receiver responds only with corresponding control packets. We assume that all parties remain in communication range of each other and that the sender and receiver are able to communicate with low error under benign conditions.

The attacker in our model performs two functions: observing and jamming. In the jamming role, the attacker can choose from a variety of jamming strategies and corresponding parameters. In the observing role, the attacker infers performance characteristics of the sender and receiver. In order to formulate the jamming attack model of interest, we first present a mathematical model for this system from the perspective of the attacker. The notation used for the remainder of this chapter is given in Table 6.1.

Although the signals transmitted and received by the attacker are continuous, the packet observations and attack decisions are discrete-time events. We thus define the system model, as viewed by the attacker, in terms of a discrete time step k, where the length of k can be periodic or event driven.

Table 6.1: We provide a summary of the notation used through the remainder of this chapter.

Definitions	
k	Discrete time variable
$u_k(t)$	Signal broadcast by the jammer in time step k
$w_k(t)$	Signal observed by the jammer in time step k
$\mathcal{H}$	System transfer function
$\boldsymbol{\phi}_k$	Performance parameters for time step k
$\mathcal{S}$	Jamming strategy
$\boldsymbol{p}_k$	Jammer parameters for k^{th} time step
$\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$	Jamming metrics for strategy $\mathcal{S}$
$\iota(\mathcal{S}, \boldsymbol{p})$	Jamming metric for impact
$\varsigma(\mathcal{S}, \boldsymbol{p})$	Jamming metric for stealth
$\eta(\mathcal{S}, \boldsymbol{p})$	Jamming metric for expenditure
Π_k	Packet delivery ratio for time step k
P_{det}	Probability of attack detection
$\mathcal{G}$	Discrete mapping of $\boldsymbol{p} \mapsto \boldsymbol{\phi}$
ϵ	Error in estimate of $\boldsymbol{\phi}$
μ	Normalized combination of jamming metrics
T	Target value for tuning based STIR-jamming

We define the continuous signal that the attacker broadcasts during time step k as $u_k(t)$ and the continuous signal observed by the attacker during time step k as $w_k(t)$. To capture the relationship between the jamming signal $u_k(t)$ and the jammer-influenced observation $w_k(t)$, we define the transfer function $\mathcal{H}$. In terms of the model in Figure 6.1, $\mathcal{H}$ replaces the sender, receiver, and the channels between the three parties, yielding the jammer-centric mathematical model shown in Figure 6.2.

The continuous signal $w_k(t)$ is composed of both control and data packet communication between the sender and receiver. The observed communications are under the influence of both noise and fading over the wireless sender-to-attacker and receiver-to-attacker channels respectively. In order to facilitate the discrete decision process of the attacker, we suppose that the attacker aggregates and summarizes the time domain signals into a vector $\boldsymbol{\phi}_k$, which represents an observation of a set of sender-to-receiver performance metrics of interest. We further discuss the mapping between the observed signal $w_k(t)$ and the summary $\boldsymbol{\phi}_k$ in Section 6.2.1.

We define $u_k(t)$ as the continuous signal that the jammer transmits using a mapping from

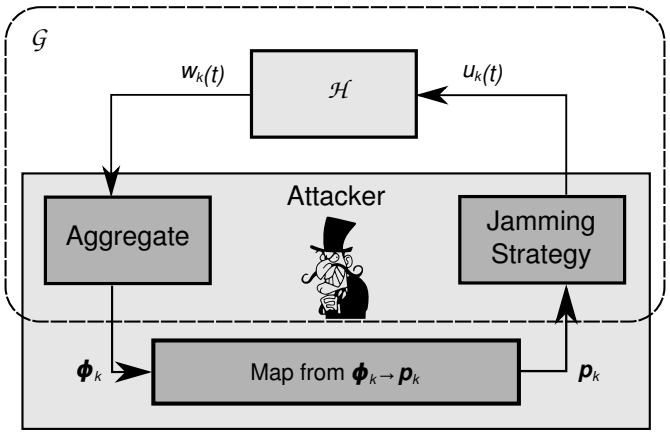


Figure 6.2: In our system model, the attacker generates a jamming signal $u_{k+1}(t)$ in response to previously observed signals $w_k(t)$ from the target system, represented by a transfer function $\mathcal{H}$. We further abstract this model to relate the jammers parameter selection $\boldsymbol{p}_k$ to the observed signal $\boldsymbol{\phi}_k$.

the discrete parameter space of the jammer to the actual jamming signal. We define a jamming strategy function $\mathcal{S}$ and a jamming parameter vector $\boldsymbol{p}_k$, such that $u_k(t) = \mathcal{S}(\boldsymbol{p}_k)$. The strategy $\mathcal{S}$ defines the type of jamming attack being mounted, and the parameter vector $\boldsymbol{p}_k$ specifies a number of parameters that are left free by the general strategy.

Similar to the continuous transfer function $\mathcal{H}$ that maps $u_k(t) \mapsto w_k(t)$, we define a discrete representation $\mathcal{G}$ of the transfer function that virtually maps the jamming strategy $\mathcal{S}$ and parameter vector $\boldsymbol{p}_k$ to the observation summary $\boldsymbol{\phi}_k$. The ability for the attacker to dynamically modify its attack is thus determined by choosing $\boldsymbol{p}_k$ according to estimates of $\mathcal{G}$ and observations $\boldsymbol{\phi}_k$.

6.2 STIR-Jamming

Using the system model given in Section 6.1, we next introduce *STIR-jamming* or self-tuned, inference-based, real-time jamming, in which the attacker observes the sender's and receiver's communications to continually modify attack parameters with the aim of mounting high-efficiency jamming attacks. We describe the actions of STIR-jamming as follows.

1. **Observe**: From the observed signal $w_k(t)$, create a summary of the performance metrics $\boldsymbol{\phi}_k$.
2. **Estimate**: Given the previously chosen parameters $\boldsymbol{p}_k$ and the observation $\boldsymbol{\phi}_k$, characterize the effect of the attack efforts.
3. **Optimize**: Use the estimate above to select parameters $\boldsymbol{p}_{k+1}$ according to a given collection of jamming metrics $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$.

Before presenting the algorithms in Section 6.2.3, we describe the estimation of performance characteristics $\boldsymbol{\phi}_k$ in Section 6.2.1 and selection of jamming metrics $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$ in Section 6.2.2.

6.2.1 Observation

In the observations phase of STIR-jamming the attacker converts an observed signal $w_k(t)$ into performance parameters $\boldsymbol{\phi}_k$. There are two major considerations for observation, what information is desired and how to obtain that information. What information is desired depends dramatically on the goal of the attacker. In this work, we focus on attacking a single link so we consider packet delivery ratio (PDR) as a metric.

The second question about the observation of a legitimate system is how to obtain this information. To do this an attacker can observe packet transmissions only when it is not jamming, and use statistical analysis to estimate the PDR metric. Consideration of appropriate statistical techniques and the resulting estimation accuracy is left as future work.

6.2.2 Jamming Metrics

As previously defined, the jamming metrics in the set $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$ are used to gauge the effectiveness of a jamming strategy $\mathcal{S}$ with parameters $\boldsymbol{p}$. We consider three jamming metrics that are important for an effective jamming attack. These metrics are *impact*, *stealth*, and *expenditure*, respectively denoted as $\iota(\mathcal{S}, \boldsymbol{p})$, $\varsigma(\mathcal{S}, \boldsymbol{p})$, and $\eta(\mathcal{S}, \boldsymbol{p})$.

Impact

We define impact $\iota(\mathcal{S}, \boldsymbol{p})$ as the amount of degradation a jammer causes in the sender-receiver system. One measure of impact is reduction of the PDR, also used as a performance metric in Section 6.2.1. In other words, a lower PDR indicates a higher impact.

Stealth

We define stealth $\varsigma(\mathcal{S}, \boldsymbol{p})$ as the ability of a jammer to evade detection by the sender-receiver communication system. Stealth is important because if an attack is detected, the legitimate nodes can take appropriate actions to compensate for the attack impacts or directly penalize the attacker. We propose using a stealth metric that is inversely proportional to the probability of detection P_{det}.

Expenditure

We define expenditure $\eta(\mathcal{S}, \boldsymbol{p})$ as the amount of resources the attacker uses to mount an attack. Many metrics can be considered for the resource usage in expenditure (e.g., bandwidth, energy, broadcast time, etc) but we focus specifically on average power in this work, supposing that the attacker could be a battery-powered mobile device or sensor node. Average power is a straightforward metric defined in terms of the amount of time spend jamming and the jamming signal power and can be used as an indication of energy consumption.

6.2.3 Attack Algorithm

In this section, we present our attack algorithm which uses the observation methods presented in Section 6.2.1 to optimize the jamming metrics in Section 6.2.2. We consider estimation and optimization with two approaches. The first approach is *model-based* or *mSTIR-jamming*, which uses a non-linear reference model and rigorous optimization methods. The second approach is *tuning-based* or *tSTIR-jamming*, which uses a technique of adjusting jamming parameters

to increase or decrease the attack's effect without the need for designing reference model. In what follows, we present jamming algorithms for these two approaches and a comparison of the resulting attacks.

mSTIR-jamming

The mSTIR-jamming algorithm involves the three steps of observing the performance parameters $\boldsymbol{\phi}_k$, estimating the discrete transfer function $\mathcal{G}_{k+1}$ that is expected in the next time step, and using the estimate of $\mathcal{G}_{k+1}$ to choose the subsequent attack parameters $\boldsymbol{p}_{k+1}$. The proposed algorithm uses the observation technique described in Section 6.2.1 to compute the performance parameters $\boldsymbol{\phi}_k$ at each time step.

The algorithm relies on the attacker's ability to compute an estimate $\widehat{\mathcal{G}}_k$ of the discrete transfer function $\mathcal{G}_k$ which maps $(\mathcal{S}, \boldsymbol{p}_k) \mapsto \boldsymbol{\phi}_k$, i.e. mapping the jammer's effort to its observed effect. As part of the mSTIR-jamming algorithm, the attacker updates its estimate of the discrete transfer function, using $\boldsymbol{\phi}_k$ and $\widehat{\mathcal{G}}_k$ to estimate $\widehat{\mathcal{G}}_{k+1}$. To facilitate this process, we introduce a scalar error value ϵ_k computed using the function $\mathrm{error}(\boldsymbol{\phi}_k, \widehat{\mathcal{G}}_k, \boldsymbol{p}_k)$ as

$$\epsilon_k = \mathrm{error}(\boldsymbol{\phi}_k, \widehat{\mathcal{G}}_k, \boldsymbol{p}_k). \tag{6.1}$$

Using (6.1), we define the function for updating the transfer function estimate as

$$\widehat{\mathcal{G}}_{k+1} = \mathrm{update}(\widehat{\mathcal{G}}_k, \epsilon_k). \tag{6.2}$$

In Section 6.3.3, we show how the update function can be constructed by adding a tuning parameter into the transfer function model that can be updated to better match the expected and observed performance metrics.

The third step in the mSTIR-jamming attack is to choose the attack parameters to optimize the desired jamming attack metrics. This requires a decision of how to jointly optimize the jamming

metrics in the set $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$. Toward this end, we define the optimization objective function $\mu(\mathcal{S}, \boldsymbol{p}, \widehat{\mathcal{G}}_{k+1})$ as the combination of metrics in $\boldsymbol{M}(\mathcal{S}, \boldsymbol{p})$ the attacker will aim to maximize. Assuming the existence of lower and upper bounds on the jamming parameters $\boldsymbol{p}$, denoted by $\boldsymbol{p}_{min}$ and $\boldsymbol{p}_{max}$, respectively, we define the optimization problem as

$$\begin{aligned} \underset{\boldsymbol{p}}{\text{maximize}} \quad & \mu(\mathcal{S}, \boldsymbol{p}, \widehat{\mathcal{G}}_{k+1}) \\ \text{subject to} \quad & \boldsymbol{p}_{min} \leq \boldsymbol{p} \leq \boldsymbol{p}_{max}. \end{aligned} \tag{6.3}$$

One of the benefits of mSTIR-jamming is that it does not require a perfect model of the system and parameters to be efficient. In this chapter, we use a simple reference model based off of Friis equation to optimize and, as shown in section 6.4, obtain good results. We anticipate that in future work it is possible to use universal approximators [111] and obtain good results or include context awareness to make the system adaptation even better.

tSTIR-jamming

The mSTIR-jamming attack is effective in optimizing the jamming parameters, but relies on availability of a usable approximation of the system $\mathcal{G}$. In many cases, estimating $\mathcal{G}$ with sufficient accuracy may be prohibitively costly or even infeasible. Hence, we present tSTIR-jamming to eliminate the need for this expensive estimation step.

tSTIR-jamming uses the same observation method proposed in Section 6.2.1. We define a target value $\boldsymbol{T}$ which serves as the desired value for $\boldsymbol{\phi}$. We then measure the error ϵ_k between the observation and the target as

$$\epsilon_k = \text{error}(\boldsymbol{\phi}_k, \boldsymbol{T}), \tag{6.4}$$

where a positive ϵ_k indicates that the attack was too aggressive and a negative value indicates the attack was not aggressive enough. We define the decision variable $\delta_k \in \{-1, 0, 1\}$ at time step k to indicate whether the attack effort should increase, remain unchanged, or decrease for

the subsequent time step. We let ρ indicate the threshold at which the attack should be changed, yielding

$$\delta_k = \begin{cases} -1, & \text{if } \epsilon_k \leq -\rho \\ 1, & \text{if } \epsilon_k \geq \rho \\ 0, & \text{else.} \end{cases} \tag{6.5}$$

We use an intuitive update algorithm for the tSTIR-jamming attack. If $\delta_k = 0$, then no change to the attack parameters is required, so $\boldsymbol{p}_{k+1} = \boldsymbol{p}_k$. If δ_k is non-zero, the attack effort is decreased or increased accordingly. To modify the attack parameters, we use a one-step transition in any one of the parameters in $\boldsymbol{p}_k$, effectively taking a one-dimensional step in the parameter space. Letting $\boldsymbol{p}_k^+$ and $\boldsymbol{p}_k^-$ be the sets of all possible one-step parameter vectors with increased and decreased attack impact $\iota(\mathcal{S}, \boldsymbol{p}_k)$, respectively, the next parameters $\boldsymbol{p}_{k+1}$ are chosen from the corresponding one-step parameter space using the expenditure metric $\eta(\mathcal{S}, \boldsymbol{p}_k)$.

In general, the resulting conditional optimization problem is thus given by

$$\begin{aligned} &\text{if } \delta_k = 0 \\ &\quad \boldsymbol{p}_{k+1} = \boldsymbol{p}_k \\ &\text{else if } \delta_k = 1 \\ &\quad \boldsymbol{p}_{k+1} = \underset{\boldsymbol{p} \in \boldsymbol{p}_k^-}{\arg\min}\ \eta(\mathcal{S}, \boldsymbol{p}) \\ &\text{else if } \delta_k = -1 \\ &\quad \boldsymbol{p}_{k+1} = \underset{\boldsymbol{p} \in \boldsymbol{p}_k^+}{\arg\min}\ \eta(\mathcal{S}, \boldsymbol{p}) \end{aligned} \tag{6.6}$$

A two-parameter example of the one-step transitions used in the tSTIR-jamming attack is illustrated in Figure 6.3 for two parameters p_1 and p_2. If $\boldsymbol{p}_k = (p_1, p_2)$ and $\delta_k = -1$, the attacker can choose $\boldsymbol{p}_{k+1} = (p_1 + \Delta_1, p_2)$ or $\boldsymbol{p}_{k+1} = (p_1, p_2 + \Delta_2)$, where Δ_1 and Δ_2 are pre-determined step sizes for each parameter, depending on which results in lower energy expenditure. A similar state-transition diagram can be envisioned in a higher-dimensional space for arbitrary parameters

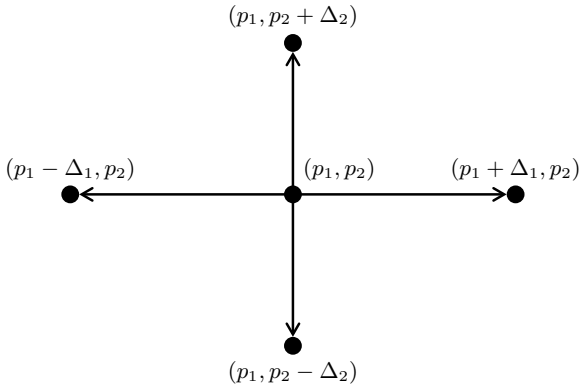


Figure 6.3: State transition rule. At the beginning of each time step, the attacker is operating at a point (p_1, p_2) in the state space; this is the center dot in the diagram. For the next time step, it can remain where it is (holding both p_1 and p_2 constant) or move to one of the other four points (incrementing or decrementing p_1 or p_2, but not both).

$\boldsymbol{p}$.

Comparison

These two algorithms both have advantages and disadvantages. mSTIR-jamming is able to estimate information on stealth which can be advantageous in a stealth-critical situation. The cost for using the model-based attack, however, is a large increase in computation because the optimization problems in mSTIR-jamming are non-trivial optimization problems for attackers with little computational power. tSTIR-jamming is not able to cope with stealth due to relaxation of the model, but it is more computationally efficient. Ultimately the choice of algorithm depends on the availability of a reasonable reference model, the need to predict stealth, and the associated computation overhead.

6.3 Case Study: IEEE 802.15.4

In this section, we use the STIR-jamming framework presented in Section 6.2 to design a proof of concept for both model-based and tuning-based STIR-jamming attacks targeting a sender-receiver system using the IEEE 802.15.4 protocol [112]. 802.15.4 is a popular low-power per-

sonal area network protocol used in many sensor network platforms. The implementations suggested in this section are just proof-of-concepts and not optimally designed attacks, we leave further research into optimal controller design for STIR-jamming as future work.

6.3.1 IEEE 802.15.4 overview

The 802.15.4 protocol is a low-power personal area network protocol commonly used in wireless sensor network platforms. In this chapter, we are primarily concerned with the 2.4 GHz physical layer protocol defined in the 802.15.4 specification [112]. This protocol uses direct sequence spread spectrum to provide robust transmission against noise and interference. The use of DSSS maps 4-bit symbols to 32-chip quasi-orthogonal patterns that are then sent at a data rate of 2 Mcps (mega-chips per second), yielding an effective data rate of 250 kbps.

We note that while the use of DSSS provides strong chip-error correction capability at the symbol level, there is no packet level error correction in the 802.15.4 standard. Instead, a two-byte checksum is included in each packet to allow the receiver to detect, with high probability, if the packet is received in error but does not allow for error correction.

6.3.2 Short Form Periodic Jamming

Short Form Periodic Jamming (SFPJ) is an efficient class of jamming attacks against the 802.15.4 protocol described in Section 6.3.1 which can be tuned to have high impact with low cost [113].

In a SFPJ attack, the jamming signal is cycled on and off with a period on the order of the symbol duration, as compared to the order of the packet duration as proposed in previous work [113]. A periodic jamming attack can be defined in terms of three jamming parameters: period, defined as the number of symbols per on-off cycle; duty cycle, defined as the percent of time the jammer is on; and amplitude, defined as the jamming signal power. It can be shown that it is possible to effectively jam an 802.15.4 link with a small duty cycle. It has been shown that SFPJ can interfere with 95% of 802.15.4 communications with a duty cycle under 10% [113].

SFPJ Metrics against 802.15.4

As discussed in Section 6.2.2, STIR-jamming is largely dependent on jamming metric definitions.Here we formulate specific instances of the *impact*, *expenditure*, and *stealth* metrics for the 802.15.4 network scenario that are used for both mSTIR-jamming and tSTIR-jamming attacks. We define our metrics in terms of the jamming signal amplitude a and duty cycle d, holding the signal period constant.

Impact: We consider packet delivery ratio (PDR) as the measure of impact on the IEEE 802.15.4 system. To do this we estimate the PDR as a function of the amplitude a and duty cycle d. Because no error correction is used in the 802.15.4 protocol, the PDR $\Pi(a, d)$ can be estimated as

$$\Pi(a, d) = (1 - P_s(a))^{nd}(1 - P_s(0))^{n(1-d)}, \tag{6.7}$$

where $P_s(x)$ is the probability of symbol error with jamming signal average power x and n is the number of symbols per packet. For the 802.15.4 protocol, the symbol error $P_s(x)$ can be further decomposed as

$$P_s(x) = \sum_{i=17}^{32} \binom{32}{i} P_c(x)^i (1 - P_c(x))^{32-i}, \tag{6.8}$$

where $P_c(x)$ is the probability of chip error for 802.15.4 under attack with a jamming signal amplitude x. This probability of chip error can be estimated as

$$P_c(x) = \frac{1}{2} Q\left(\sqrt{\frac{T_c F(d_{tr}) S_{tx} - T_c F(d_{jr}) x}{N_0}}\right), \tag{6.9}$$

where T_c is the chip duration, d_{tr} and d_{jr} are the respective transmitter-to-receiver and jammer-to-receiver distances, S_{tx} is the average signal power from the transmitter, N_0 is the ambient noise power, and $F(d)$ is a model for path loss. As long as the relative geometry (i.e., d_{tr} and d_{jr}) and the transmit power S_{tx} are known to the attacker, the PDR estimate in (6.7) can be used

to define the impact metric as

$$\iota(\mathcal{S}, \boldsymbol{p}) = 1 - \Pi(a, d), \tag{6.10}$$

where $\boldsymbol{p} = (a, d)$ as previously described.

Stealth: We consider the use of a combination of PDR and estimated signal strength S_{rx} at the receiver to measure stealth $\varsigma(\mathcal{S}, \boldsymbol{p})$. For $\boldsymbol{p} = (a, d)$ as above, the jammer can estimate the received signal strength $S_{rx}(a, d)$ as

$$S_{rx}(a, d) = S_{tx}F(d_{tr}) + adF(d_{jr}) \tag{6.11}$$

on average. The jammer can then estimate the probability P_{det} of being detected as

$$P_{det}(a, d) = \left(1 + e^{-(S_{rx}(a,d) - \kappa\Pi(a,d))}\right)^{-1}, \tag{6.12}$$

where κ is a scaling parameter that determines an acceptable threshold to relate the acceptable PDR for a given received signal strength. We provide our derivation for this equation in our previous work [18]. The estimation of the probability of detection at another node is still an open problem so when better methods are discovered they can be used in place of (6.12). The detection probability can then be used to define the stealth metric as

$$\varsigma(\mathcal{S}, \boldsymbol{p}) = 1 - P_{det}(a, d), \tag{6.13}$$

where $\boldsymbol{p} = (a, d)$.

Expenditure: As previously discussed, we measure expenditure in terms of the energy of the jamming signal. This energy expenditure can be measured directly by the jamming device as the combination of signal amplitude and duty cycle as

$$\eta(\mathcal{S}, \boldsymbol{p}) = ad \tag{6.14}$$

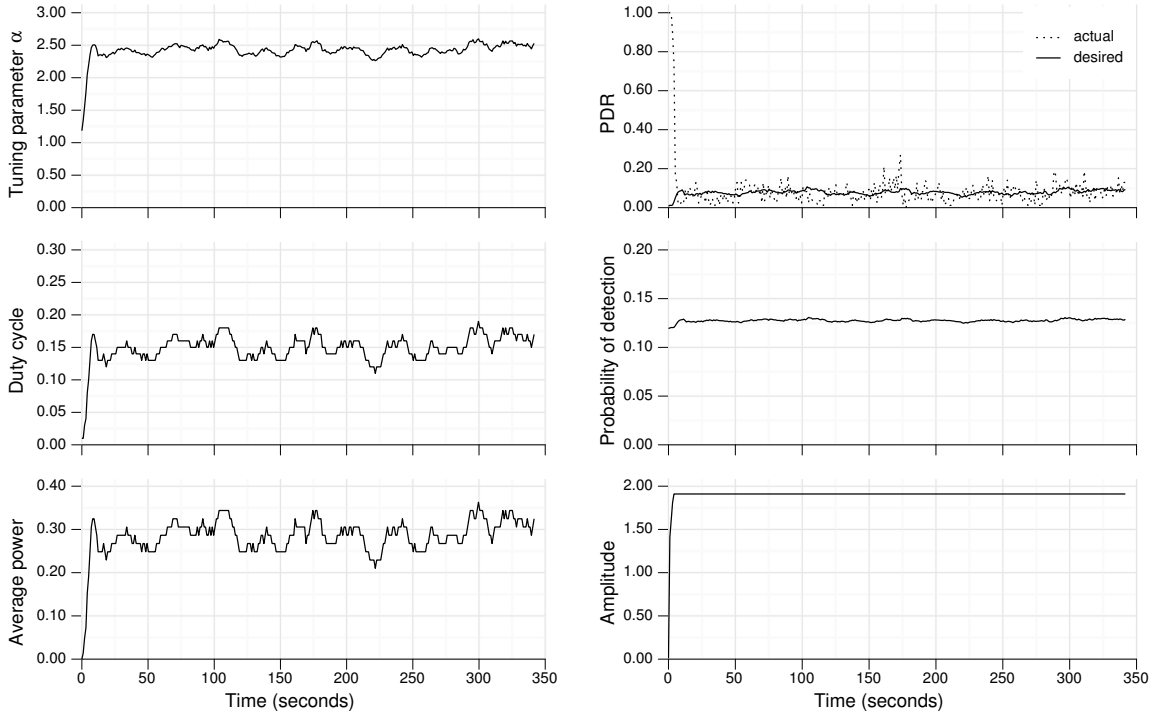


Figure 6.4: The results for the mSTIR-jamming attack are shown, with $\beta_\iota = \beta_\varsigma = \beta_\eta = 50\%$ and induced measurement error $\xi = 10\%$.

where $\boldsymbol{p} = (a, d)$.

6.3.3 mSTIR-Jamming Design

In this section, we present an instance of the mSTIR-jamming attack presented in Section 6.2.3 using the metrics given in Section 6.3.2 for a periodic jamming attack on an 802.15.4 system. As in Section 6.2, we break our attack description into the steps of observing, estimating, and optimizing.

Observation

As previously described in Section 6.2.1, we use the PDR as the observation metric to gauge the actual effect of the jamming attack on the sender-receiver system. At every time step k, the jammer thus observes a PDR value Π_k.

Estimation

The estimation step for the mSTIR-jamming attack relies on the ability for the attacker to estimate the transfer function $\mathcal{G}_k$ in each time step. We present an estimation method based on the same metrics used in Section 6.3.2, with a slight modifications to allow for adaptation. Accepting the fact that the physical layer model assumed in the computation of the chip error probability in (6.9) is not a perfect model, we introduce an additional parameter α_k to assist in fitting the model to the observations made by the attacker. We thus define the modified chip error, symbol error, and packet delivery probabilities respectively as

$$P_c^{\alpha}(x) = \frac{1}{2} Q\left(\sqrt{\frac{\alpha T_c F(d_{tr}) S_{tx} - T_c F(d_{jr}) x}{N_0}}\right), \tag{6.15}$$

$$P_s^{\alpha}(x) = \sum_{i=17}^{32} \binom{32}{i} P_c^{\alpha}(x)^i (1 - P_c^{\alpha}(x))^{32-i}, \tag{6.16}$$

$$\Pi^{\alpha}(a, d) = (1 - P_s^{\alpha}(a))^{nd} (1 - P_s^{\alpha}(0))^{n(1-d)}. \tag{6.17}$$

The estimated transfer function $\hat{\mathcal{G}}_k$ mapping $\boldsymbol{p}_k = (a_k, d_k)$ to $\boldsymbol{\phi}_k = \Pi^{\alpha_k}(a_k, d_k)$ is thus provided by (6.15), (6.16), and (6.17), parameterized by the tuning variable α_k which is updated at each time step.

Using this estimation model in the mSTIR-jamming attack algorithm to compute an estimate $\hat{\mathcal{G}}_k$ via α_k then allows for the error between the predicted PDR $\Pi^{\alpha_k}(a_k, d_k)$ and the observed PDR Π_k at time step k to be defined as

$$\text{error}\left(\Pi_k, \Pi^{\alpha_k}(a_k, d_k), (a_k, d_k)\right) = \Pi_k - \Pi^{\alpha_k}(a_k, d_k). \tag{6.18}$$

Using the above model, we also define the update function via α_{k+1} as a function of the

previous α_k and ϵ_k as

$$\alpha_{k+1} = \begin{cases} \alpha_k(1+\epsilon_k), & \text{if } \epsilon_k > 0 \\ \alpha_k(1-\epsilon_k)^{-1}, & \text{otherwise.} \end{cases} \tag{6.19}$$

Optimization

Lastly, we present the optimization formulation used for the mSTIR-jamming. In choosing $\boldsymbol{p}_{k+1} = (a_{k+1}, d_{k+1})$, the algorithm imposes lower bound $\boldsymbol{p}_{min} = (a_{min}, d_{min})$ and upper bound $\boldsymbol{p}_{max} = (a_{max}, d_{max})$, where $a_{min} = 0$, $a_{max} > 0$ is the maximum jamming power, and d_{min} and d_{max} are specified bounds on the duty cycle (satisfying $0 \leq d_{min} < d_{max} \leq 1$). The objective function $\mu(\mathcal{S}, \boldsymbol{p}, \hat{\mathcal{G}}_{k+1}) = \mu(a, d)$ can be any combination of the metrics in $\boldsymbol{M}(\mathcal{S})$. We choose a linear combination of the impact, stealth, and expenditure metrics defined in Section 6.2.2, modified for use with the α_k parameterization, as

$$\mu(a, d) = \beta_\iota \iota^{\alpha_{k+1}}(a, d) - \beta_\eta \eta^{\alpha_{k+1}}(a, d) + \beta_\varsigma \varsigma^{\alpha_{k+1}}(a, d), \tag{6.20}$$

where β_ι, β_η, and β_ς are scalar weights to indicate the attacker's preference and to scale the metrics into comparable ranges. We further discuss the β parameters in Section 6.4.

6.3.4 tSTIR-Jamming Design

We next present an instance of the tSTIR-jamming attack presented in Section 6.2.3 using the metrics in Section 6.3.2 for a similar periodic jamming attack on an 802.15.4 system.

Similar to the case of the model-based attack, we use the PDR as the observation metric of interest to measure the impact of the jamming attack on the sender-receiver system, again through the observation of a PDR value Π_k at each time step. In the tuning-based attack, the estimation and optimization steps are computationally simpler than in the model-based attack. Given the target value T as the PDR desired by the attacker, the error between the observed PDR Π_k and

Figure 6.5: In this figure, we show the SDR setup we used for testing the STIR-jamming algorithms.

the target T is given by

$$\text{error}(\Pi_k, T) = \Pi_k - T. \tag{6.21}$$

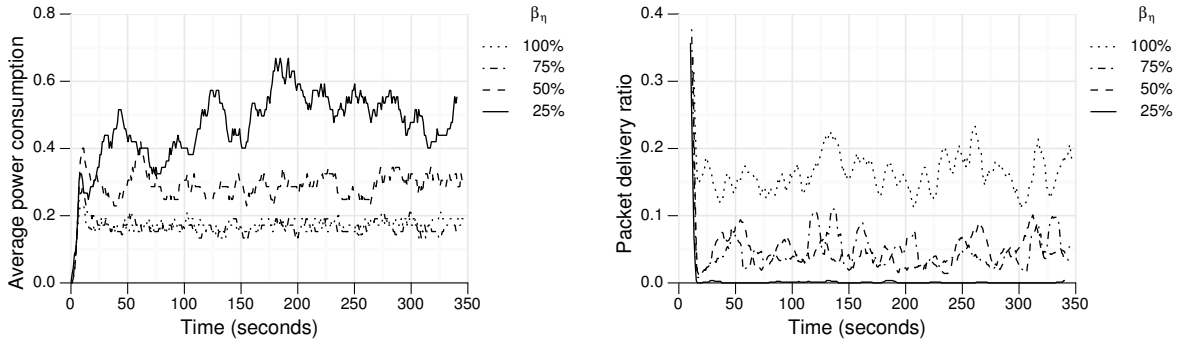
For a given value of ρ, the δ_k decision parameter then allows for the one-step transition to be made from the previous parameters (a_k, d_k) to the subsequent parameters (a_{k+1}, d_{k+1}). In this case, the one-step transitions follow the logic given in the example in Figure 6.3.

6.4 Implementation Results

In this section, we discuss our proof-of-concept implementation of mSTIR-jamming and tSTIR-jamming attacks described in Section 6.3 and present our performance results. The implementation of both algorithms uses the USRP2 software-defined radio platform [114] with the GNU Radio software package [115]. We show the physical setup we use for testing in Figure 6.5. We use a previously developed implementation of the 802.15.4 protocol from UCLA [116], and we develop our customized jamming attack mechanisms to implement the attacks. We present the results individually and then provide a brief comparison of the two sets of results.

6.4.1 mSTIR-Jamming Results

We implemented the mSTIR-jamming attack as described in Section 6.3.3. As previously mentioned, the estimation process involved in observing the PDR Π_k in each time step, so we instead provide this statistic to the attacker via a direct line from the receiver. However, to test the per-



(a) (b)

Figure 6.6: In (a) we show the average power consumed by the jammer, for four possible values of the expenditure parameter β_η, with β_ι and β_ς fixed at 50%. In this figure, higher values of β_η cause the jammer to reduce power consumption. In (b) We show the PDR achieved by the jammer, with the same settings as (a).

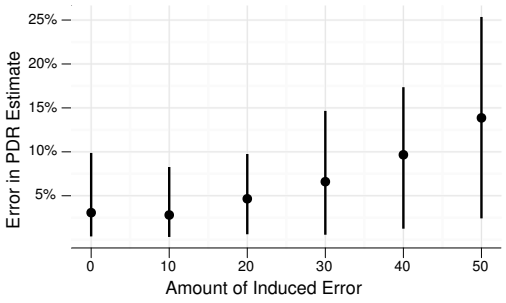


Figure 6.7: The percent error in the attacker's estimated PDR compared to the actual PDR at the receiver is plotted as a function of the level of error ξ induced in the PDR measurements. The error bars show one standard deviation around the mean (with a minimum of zero).

formance of our STIR-jamming formulation with various levels of error, we induce errors into the observed PDR Π_k. In particular, the observation Π_k given to the attacker is equal to the true PDR perturbed by a uniform random variable in the interval $[-\xi, \xi]$, and we test several values of ξ in our implementation.

The results of one trial run of the mSTIR-jamming attack are shown in Figure 6.4. From the right-hand column, it can be seen that the jamming amplitude, probability of detection, and PDR goal rapidly stabilize; the actual achieved PDR fluctuates on a moment-to-moment basis but rarely exceeds 0.2, marking a successful attack. Probability of detection is also consistently low. The left-hand column shows the time evolution of the tuning parameter α, which does jitter

a bit but is stable overall, and the duty cycle and average power, which track α precisely (the apparent additional variation is an artifact of the scale).

Figure 6.6(a) and Figure 6.6(b) demonstrate an STIR-jamming scenario where the jamming metrics are not equally weighted in (6.20). In both of these figures, the weights for the impact and stealth metrics are fixed at $\beta_\iota = \beta_\varsigma = 50\%$, while the weight for the expenditure metric is varied among $\beta_\eta = 25\%, 50\%, 75\%, 100\%$ during the four tests. When the expenditure is weighted heavily ($\beta_\eta = 75\%$ and $\beta_\eta = 100\%$), the resulting signal power and the corresponding jamming impact are both reduced. When the expenditure is lightly weighted ($\beta_\eta = 50\%$), the attack requires significantly higher resources but also yields significantly higher impact (i.e., lower PDR). We note that the attacker could also adapt these weights depending on energy availability (i.e., system health), change in jamming goals, or other dynamics, though such weight modification is beyond the scope of this work.

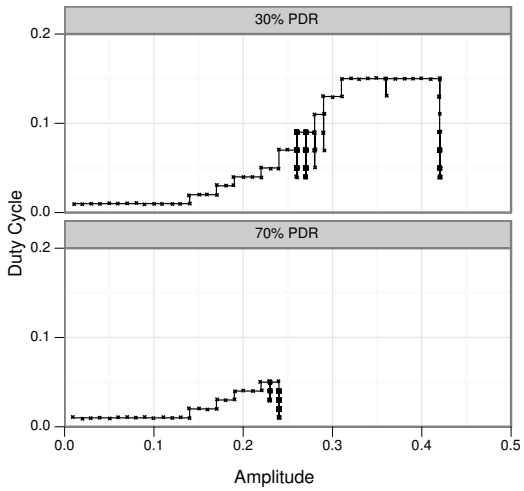


Figure 6.8: The time evolution of control parameters for the tSTIR-jamming attack is shown, with two different PDR targets. The boxes are jittered slightly to reveal where the search stabilized: denser blobs indicate longer dwell times.

6.4.2 tSTIR-Jamming Results

We also demonstrate the effect of the induced error in the PDR measurements taken by the attacker. Figure 6.7 shows the error in PDR estimation as a function of the level of error ξ induced in the measurement. The worst case ($\xi = 50\%$) parameter tested demonstrates that the measurement error does affect the result but does not defeat the STIR-jamming attack.

We implemented the tSTIR-jamming attack as described in Section 6.3.4. Figure 6.8 illustrates the parameter trajectory through the state space for two target values, $T = 30\%$ PDR and $T = 70\%$ PDR. In both cases, we see that the attack parameters stabilize relatively quickly to a small region of the state space, although in the case of $T = 30\%$ PDR, the parameters jump to a different region after a short time.

The results of the tSTIR-jamming attack are shown in detail in Figure 6.9. This attacker can hold the system under attack to within about 15% of the PDR goal, although with a great deal of variance. Though this is generally undesirable for control systems, it may actually benefit the attacker in terms of reducing the chance of detection.

The plot of signal power in Figure 6.9 also sheds some light on the jump in the state space that was observed in Figure 6.8. After about 500 seconds during the $T = 30\%$ trial, something changed in the experimental environment (possibly a reduction of external network use) that allowed the sender-receiver system to improve packet delivery. The attacker reacted to this external event by first increasing its duty cycle and then increasing the signal power. After only a few seconds, the attacker adapted to the changes and drove the PDR back to the desired target. This ability to adapt to dynamic environmental conditions is one of the major advantages of STIR-jamming attacks compared to using static parameters.

Comparing the mSTIR-jamming and tSTIR-jamming attacks is interesting, but since the trials were taken during different times of day with slightly different hardware configuration a direct comparison is not possible at this time. However, one interesting point of comparison is that the tuning-based attack uses considerably less power. Since the model-based attack re-optimizes

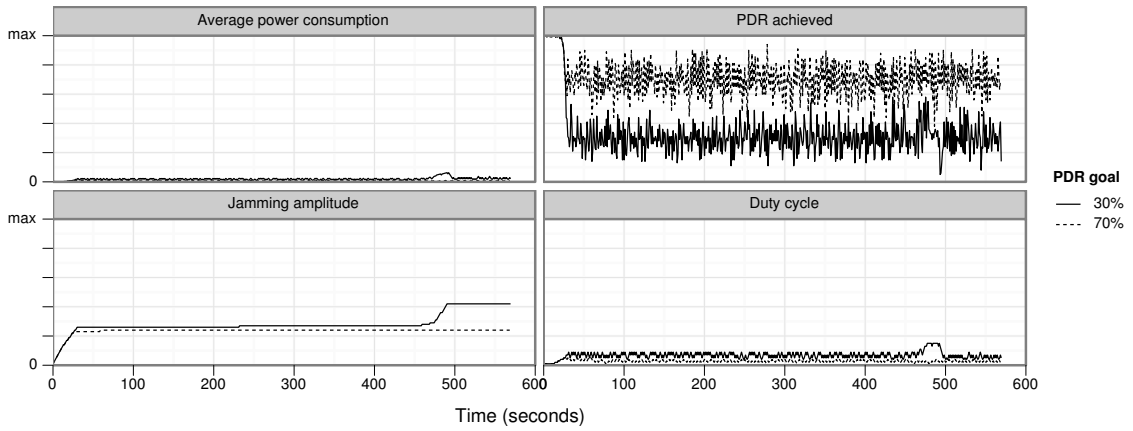


Figure 6.9: The performance of a tSTIR-jammer is shown for a goal of 30% and 70%.

the power level and duty cycle at every time step, it often leads to very loud pulses with power near the maximum, whereas the tuning-based attack slowly changes the power level only when the benefit of doing so is sufficiently large. Another trend we observed is that the model-based attack tends to be more stable than the tuning-based attack, as the estimate of the system transfer function converges relatively quickly with considerably less variance. Further comparison of the different attack types and investigation of the stability of the attacks are beyond the scope of this work and are left for future consideration.

6.5 Discussion

In this chapter, we introduce a framework for adaptation in wireless network attackers. This basic framework allows an attacker to listen to and infer information about a legitimate system using commodity hardware and adapt its attack to find more robust attack models. To prove the value of this concept we introduce self-tuned, inference-based, real-time jamming or *STIR-jamming*. This attack allows for jamming attack parameters to be tuned real-time to optimize an attacker's *impact*, *stealth*, or *expenditure*. This is accomplished by continually observing the system under attack, estimating the impact of jamming, and using this information to optimize its attacks. We show two proof of concept implementations for this attack, mSTIR-jamming which uses a rough

channel model to optimize over possible attack parameters and tSTIR-jamming which searches for optimal attacks by taking small search steps in the parameter space. We implement proof of concept versions of these attacks which are able to find stable attack parameter locations to degrade an 802.15.4 links performance with high efficiency and low detectability.

This demonstrates the ability of an attacker to interact with a constant defender in a unknown system. The attacker is effectively able to learn the system parameters and tune her attack. This is shown to be done online and can be applied to a wide range of attacks with tunable parameters.

6.5.1 Limitations and Future Work

One main limitation of this work is how to infer knowledge about the system-under-attack. In our simulations, we directly feed-back this information from the receiver. In a real system this real-time feedback would not be possible. There are various ways this could be overcome. First, a full-duplex receiver can be used allowing for simultaneous transmission and reception. Second, partial observations can allow message reconstruction and extraction of valuable header information. Third, a listening period can allow for observation, allowing performance inferences based on packet numbers.

In the future, we will also consider the use of other learning methods, Particularly when considering non-constant defenders. For example, it would be more efficient to attack using a longer period when the defender is broadcasting intermittently but this might not be captured in our model based approach.

Another limitation of this work is that the defender has to be constant or slow. Our technique can adapt through defenders that slowly change parameters, possibly finding a new parameter every 10 seconds. It would be interesting to consider attackers for other types of defenders. For example, we can consider a defender that adapts based on the system's performance. In this case, an attacker would have to choose a strategy and consider stability in their adaptation.

Second, we could consider how to use this approach when a wider range of attacks are avail-

able allowing us to ”tune“ the attack we are using. For example, we could consider a set of 5-10 attacks that may be mounted against a wireless sensor network. We can adaptively choose which attack and parameters to use based on feedback and selection. This type of selection is more complex and require exploring a larger search space.

We can also consider how system identification can be done when the shape of the system model is not known. An attacker that is interacting with a complex system with little information may run into this scenario. For example, we can consider how an attacker can use universal approximators to model the system and choose optimal parameters.

We are also interested in how system identification techniques can be used for defensive applications. For example we are interested in exploring how this type of learning can be applied to multi-agent defense techniques for ad hoc networks.

Chapter 7

Conclusion

Advances in cyber-physical systems have enabled the use of agile, sensor-driven, feedback computing for a wide range of applications including defense, power, transportation, communications, and healthcare. These applications are in safety-critical and privacy-critical domains so their design must consider security. In this thesis, we explore security in a two-player adversarial system with an attacker and a defender. This allows us to explore realistic cyber-physical attack and defense scenarios. We are particularly interested in how the design of an attack or defenses changes based on the information that is known about the CPS and a player's opponent.

We consider four levels of knowledge that a player can have about the CPS they interact with. A player has full-knowledge when she knows all update equations, dynamics, and parameters of a system. A player has partial knowledge with slow-changing parameters when she knows the update equations of a system but not the parameters of the system and the parameters change slowly. Similarly, a player has partial knowledge with fast-changing parameters when she knows the update equations but not the fast-changing parameters. A player with no knowledge does not know the dynamics of the system, update equations, or parameters.

Similarly we introduce three levels of knowledge that a player can have about their opponent. If a player's opponent uses a known strategy, the player can deterministicly know their opponents play. If a player's opponent uses an assumed strategy, the player assumes that their opponent is

		System Knowledge			
		Known	Slow Parameter	Fast Parameter	Unknown
Opponent	Known	Chapter 3			
	Assumed	Chapter 5	Chapter 6		
	Unknown	Chapter 4			

Table 7.1: In this table, we illustrate the broad problem space of adversarial scenarios in CPS. We highlight the problems explored in this thesis in yellow and the problems left for future work in grey.

using a certain set of rules. This can include a player that assumes their opponent is constant, game-theoretic rational, or random. Lastly, if a player's opponent uses an unknown strategy then the player knows nothing about their opponents action.

We summarize all of the system and opponent knowledge levels Table 7.1. The yellow boxes indicate the topics we explore in this thesis, as well as the chapters we consider them in. We design various attack and defense strategies for a player that leverages the knowledge available to them. In Chapter 3, we show that when an attacking player has full knowledge of the system and their opponent's strategy then optimal control techniques can be used do design devastating attacks. We demonstrate this in a platoon of cars and demonstrate three attacks including the sandwich attack. In the sandwich attack an attacker causes the three cars following her to simultaneously collide which has a devastating impact on the cars involved, and potentially following cars on the highway.

In Chapter 4, we explore how a known critical system can be defended against an unknown attacker. We consider this scenario in a platoon of cars where an attacker may or may not be. Each car uses information from many cars to model the expected behavior of the car preceding them and uses this model to detect abnormal behavior. If abnormal behavior is detected the car switches to non-cooperative adaptive cruise control, which is safe even if the preceding car is malicious.

In Chapter 5, we explore how two constrained rational player interact with a known system.

We design a two-player game using a jamming model and design a dynamic programming solution to find the optimal mixed strategy. We compare this strategy to adaptive, random, and constant strategy which highlights the trade-offs of rational strategies. For example, we find that in a bullying attack, when an attacker does nothing, a defender may respond by doing nothing since the attacker maintains a high energy level over all times.

In Chapter 6, we eliminate the assumption that the system is known but assume a constant opponent. We design a jamming attack that is able to observe the system and tune it's attack to balance energy usage, effectiveness, and stealth. We do this using two approaches including a feedback controller and a model-based parameter tuning. In both case it is shown to be effective at improving attack efficiency.

7.1 Future Work

This work demonstrates the value of opponent and system specific attack and defense strategies, but as shown in Table 7.1 there is a much larger research space. One interesting future direction would be to explore how a player can interact with a slow-changing system with a rational or adaptive opponent. This problem introduce unique challenges which can leverage interesting tools and techniques. For example, a rational player with a slow changing system could explore learning reinforced optimal game strategies.

Another interesting opportunities would be how to interact with a completely unknown system. With a constant defender, it may be possible to model the attacker's impact on the system using universal approximation techniques [117] and perturbation analysis. If the player's opponent is unknown though it is unclear what type of analysis is possible. It would also be interesting to consider designing defense mechanisms that mislead an attacker trying to learn. In the future we are interested in exploring how a defender can leverage the uncertainty to fool an attacker.

Another future direction is to explore the interaction of sub-systems in CPS. For example, there may be software-defined radio on a platoon of cars. This scenario opens up the opportunity

for communication system to adapt with the performance of the cars or vehicles to adapt based on communication system performance. This opens up opportunities for attacks that leverage multiple aspects of the system to destabilize it. Likewise, this could allow for a defender to optimize their communications for robustness based on system performance to mitigate the impact of an attack.

Bibliography

[1] Y. Mo, T.-H. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2012. 1, 2.1.3, 2.1.3

[2] R. M. Gerdes, C. Winstead, and K. Heaslip, "Cps: an efficiency-motivated attack against autonomous vehicular transportation," in *Proceedings of the 29th Annual Computer Security Applications Conference*, pp. 99–108, ACM, 2013. 1, 2.2, 4.5.1

[3] T. Parker, "Stuxnet redux: Malware attribution & lessons learned," *Black Hat DC 2011–URL: www. blackhat. com/html/bh-dc-11/bh-dc-11-archives. html# Parker*, 2011. 1

[4] G. Bell, "Moores law evolved the pc industry; bells law disrupted it with players, phones, and tablets: New platforms ,tools ,and services," tech. rep., 2014. 1.1

[5] C. McLellan, "Storage in 2014: An overview," 2014. `http://www.zdnet.com/article/storage-in-2014-an-overview/`. 1.1

[6] "Injury prevention & control: Key data and statistics." Center for Disease Control. `http://www.cdc.gov/injury/overview/data.html`, Accessed: 2015-01-18. 1.1.1

[7] A. B. Hillel, R. Lerner, D. Levi, and G. Raz, "Recent progress in road and lane detection: a survey," *Machine Vision and Applications*, vol. 25, no. 3, pp. 727–745, 2014. 1.1.1

[8] A. Mogelmose, M. M. Trivedi, and T. B. Moeslund, "Vision-based traffic sign detection and analysis for intelligent driver assistance systems: Perspectives and survey," *Intelligent*

Transportation Systems, IEEE Transactions on, vol. 13, no. 4, pp. 1484–1497, 2012. 1.1.1

[9] D. Swaroop, "String stability of interconnected systems: An application to platooning in automated highway systems," *California Partners for Advanced Transit and Highways (PATH)*, 1997. 1.1.1

[10] Chen, "Estimation of car following safety: Application to the design of intelligent cruise control," *1996, PhD Dissertation*. 1.1.1, 4.2.1

[11] M. P. Lammert, A. Duran, J. Diez, K. Burton, and A. Nicholson, "Effect of platooning on fuel consumption of class 8 vehicles over a range of speeds, following distances, and mass," tech. rep., SAE Technical Paper, 2014. 1.1.1

[12] Y. Zhao, P. Minero, and V. Gupta, "On disturbance propagation in vehicle platoon control systems," in *American Control Conference (ACC), 2012*, pp. 6041–6046, IEEE, 2012. 1.1.1

[13] W. H. Heemels, A. R. Teel, N. van de Wouw, and D. Nesic, "Networked control systems with communication constraints: Tradeoffs between transmission intervals, delays and performance," *Automatic Control, IEEE Transactions on*, vol. 55, no. 8, pp. 1781–1796, 2010. 1.1.1, 2.2, 3.1.1

[14] D. Nesic and A. R. Teel, "Input-output stability properties of networked control systems," *Automatic Control, IEEE Transactions on*, vol. 49, no. 10, pp. 1650–1667, 2004. 1.1.1, 2.2

[15] M. Tabbara and D. Nesic, "Input–output stability of networked control systems with stochastic protocols and channels," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1160–1175, 2008. 1.1.1, 2.2

[16] D. J. Torrieri, *Principles of Secure Communication Systems*. Boston: Artech House, 2nd ed., 1992. 1.1.2, 2.3.1

[17] K. Pelechrinis, M. Iliofotou, and S. Krishnamurthy, "Denial of service attacks in wireless

networks: the case of jammers," *IEEE Comm Surveys and Tutorials*, Apr. 2011. 1.1.2, 2.3

[18] B. DeBruhl, Y. Kim, Z. Weinberg, and P. Tague, "Stir-ing the wireless ether with self-tuned, inference-based, real-time jamming," in *MASS 2012, proceedings*, (Las Vegas, USA), IEEE, Oct. 2012. 1.1.2, 1.2.4, 2.3.1, 6.3.2

[19] B. DeBruhl and P. Tague, "How to jam without getting caught: Analysis and empirical study of stealthy periodic jamming," in *IEEE International Conference on Sensing, Communication, and Networking (SECON)*. to appear. 1.1.2, 2.3.1, 5, 6

[20] W. Xu, K. Ma, W. Trappe, and Y. Zhang, "Jamming sensor networks: Attack and defense strategies," *IEEE Network*, vol. 20, pp. 41–47, May/June 2006. 1.1.2, 2.3.1, 5

[21] A. Chan, X. Liu, G. Noubir, and B. Thapa, "Control channel jamming: Resilience and identification of traitors," in *Proc. IEEE International Symposium on Information Theory (ISIT'07)*, (Nice, France), June 2007. 1.1.2, 5

[22] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders, "Reactive jamming in wireless networks: How realistic is the threat?," in *Proc. 4th ACM Conference on Wireless Network Security*, (Hamburg, Germany), June 2011. 1.1.2, 5

[23] A. Richa, S. Schmid, C. Scheideler, and J. Zhang, "A jamming resistant mac protocol for multi-hop wireless networks," in *Proc. of the 24th Int.Symposium on Princ. of Distributed Computing*, 2010. 1.1.2, 5

[24] L. Lazos and M. Krunz, "Selective jamming dropping insider attacks in wireless mesh networks," *IEEE Network*, vol. 25, no. 1, pp. 30–34, 2011. 1.1.2, 5

[25] X. Liu, G. Noubir, R. Sundaram, and S. Tan, "SPREAD: Foiling smart jammers using multi-layer agility," in *26th IEEE International Conference on Computer Communications (INFOCOM'07)*, (Anchorage, AK, USA), May 2007. 1.1.2, 2.3.1, 5

[26] M. J. Osborne and A. Rubenstein, *A Course in Game Theory*. MIT Press, 1994. 1.2.3

[27] R. Rajkumar and I. Lee, "Nsf workshop on cyber-physical systems," 2006. 2.1

[28] R. A. Gupta and M.-Y. Chow, "Networked control system: overview and research trends," *Industrial Electronics, IEEE Transactions on*, vol. 57, no. 7, pp. 2527–2535, 2010. 2.1.1

[29] L. Schenato, "Optimal estimation in networked control systems subject to random delay and packet drop," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1311–1317, 2008. 2.1.1

[30] L. Zhang, Y. Shi, T. Chen, and B. Huang, "A new method for stabilization of networked control systems with random delays," *Automatic Control, IEEE Transactions on*, vol. 50, no. 8, pp. 1177–1181, 2005. 2.1.1

[31] G. Xie and L. Wang, "Stabilization of networked control systems with time-varying network-induced delay," in *Proceedings of the 43rd IEEE Conference on Decision and Control, December 14–17, 2004, Atlantis, Paradise Island, Bahamas*, 2004. 2.1.1

[32] I. Pan, S. Das, and A. Gupta, "Tuning of an optimal fuzzy pid controller with stochastic algorithms for networked control systems with random time delay," *ISA transactions*, vol. 50, no. 1, pp. 28–36, 2011. 2.1.1

[33] F. Yang, Z. Wang, Y. Hung, and M. Gani, "H control for networked systems with random communication delays," *Automatic Control, IEEE Transactions on*, vol. 51, no. 3, pp. 511–518, 2006. 2.1.1

[34] D. Yue, Q.-L. Han, and C. Peng, "State feedback controller design of networked control systems," in *Control Applications, 2004. Proceedings of the 2004 IEEE International Conference on*, vol. 1, pp. 242–247, IEEE, 2004. 2.1.1

[35] J. Nilsson, B. Bernhardsson, and B. Wittenmark, "Stochastic analysis and control of real-time systems with random time delays," *Automatica*, vol. 34, no. 1, pp. 57–64, 1998. 2.1.1

[36] Y. Tipsuwan and M.-Y. Chow, "Control methodologies in networked control systems," *Control engineering practice*, vol. 11, no. 10, pp. 1099–1111, 2003. 2.1.1

[37] M. S. Branicky, S. M. Phillips, and W. Zhang, "Stability of networked control systems:

Explicit analysis of delay," in *American Control Conference, 2000. Proceedings of the 2000*, vol. 4, pp. 2352–2357, IEEE, 2000. 2.1.1

[38] H. Shousong and Z. Qixin, "Stochastic optimal control and analysis of stability of networked control systems with long delay," *Automatica*, vol. 39, no. 11, pp. 1877–1884, 2003. 2.1.1

[39] L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, and S. S. Sastry, "Foundations of control and estimation over lossy networks," *Proceedings of the IEEE*, vol. 95, no. 1, pp. 163–187, 2007. 2.1.1

[40] B. Sinopoli, L. Schenato, M. Franceschetti, K. Poolla, and S. Sastry, "Optimal linear lqg control over lossy networks without packet acknowledgment," *Asian Journal of Control*, vol. 10, no. 1, pp. 3–13, 2008. 2.1.1

[41] E. Garone, B. Sinopoli, and A. Casavola, "Lqg control over lossy tcp-like networks with probabilistic packet acknowledgements," *International Journal of Systems, Control and Communications*, vol. 2, no. 1, pp. 55–81, 2010. 2.1.1

[42] A. Teixeira, D. Pérez, H. Sandberg, and K. H. Johansson, "Attack models and scenarios for networked control systems," in *Proceedings of the 1st international conference on High Confidence Networked Systems*, pp. 55–64, ACM, 2012. 2.1.1

[43] S. Amin, A. A. Cárdenas, and S. S. Sastry, "Safe and secure networked control systems under denial-of-service attacks," in *Hybrid Systems: Computation and Control*, pp. 31–45, Springer, 2009. 2.1.1

[44] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *Communication, Control, and Computing, 2009. Allerton 2009. 47th Annual Allerton Conference on*, pp. 911–918, IEEE, 2009. 2.1.1

[45] C. Günther, "A survey of spoofing and counter-measures," *Navigation*, vol. 61, no. 3, pp. 159–177, 2014. 2.1.1

[46] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (ddos) flooding attacks," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 2046–2069, 2013. 2.1.2

[47] M. Bhatia and V. Manral, "Summary of cryptographic authentication algorithm implementation requirements for routing protocols," 2011. 2.1.2

[48] K. Zeng, K. Govindan, and P. Mohapatra, "Non-cryptographic authentication and identification in wireless networks," *network security*, vol. 1, p. 3, 2010. 2.1.2

[49] T. Aura, "Strategies against replay attacks," in *Computer Security Foundations Workshop, 1997. Proceedings., 10th*, pp. 59–68, IEEE, 1997. 2.1.2

[50] M. Zhu and S. Martinez, "On resilient consensus against replay attacks in operator-vehicle networks," in *American Control Conference (ACC), 2012*, pp. 3553–3558, IEEE, 2012. 2.1.2

[51] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems*, pp. 162–175, ACM, 2004. 2.1.2

[52] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad hoc networks*, vol. 1, no. 2, pp. 293–315, 2003. 2.1.2

[53] D. Kundur, X. Feng, S. Mashayekh, S. Liu, T. Zourntos, and K. L. Butler-Purry, "Towards modelling the impact of cyber attacks on a smart grid," *International Journal of Security and Networks*, vol. 6, no. 1, pp. 2–13, 2011. 2.1.3

[54] A. Giani, E. Bitar, M. Garcia, M. McQueen, P. Khargonekar, and K. Poolla, "Smart grid data integrity attacks: characterizations and countermeasures π," in *Smart Grid Communications (SmartGridComm), 2011 IEEE International Conference on*, pp. 232–237, IEEE, 2011. 2.1.3

[55] A. Banerjee, K. K. Venkatasubramanian, T. Mukherjee, and S. K. Gupta, "Ensuring safety,

security, and sustainability of mission-critical cyber–physical systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 283–299, 2012. 2.1.3

[56] A. Banerjee, K. Venkatasubramanian, and S. K. Gupta, "Challenges of implementing cyber-physical security solutions in body area networks," in *Proceedings of the Fourth International Conference on Body Area Networks*, p. 18, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2009. 2.1.3

[57] D. Swaroop, J. Hedrick, C. Chien, and P. Ioannou, "A comparision of spacing and headway control laws for automatically controlled vehicles1," *Vehicle System Dynamics*, vol. 23, no. 1, pp. 597–625, 1994. 2.2

[58] P. Y. Li and A. Shrivastava, "Traffic flow stability induced by constant time headway policy for adaptive cruise control vehicles," *Transportation Research Part C: Emerging Technologies*, vol. 10, no. 4, pp. 275–301, 2002. 2.2

[59] J. Ploeg, B. T. Scheepers, E. van Nunen, N. van de Wouw, and H. Nijmeijer, "Design and experimental evaluation of cooperative adaptive cruise control," in *Intelligent Transportation Systems (ITSC), 2011 14th International IEEE Conference on*, pp. 260–265, IEEE, 2011. 2.2, 3.1.1

[60] F. Lin, M. Fardad, and M. R. Jovanovic, "Optimal control of vehicular formations with nearest neighbor interactions," *Automatic Control, IEEE Transactions on*, vol. 57, no. 9, pp. 2203–2218, 2012. 2.2, 3.1

[61] R. Teo, D. Stipanovic, and C. Tomlin, "Decentralized spacing control of a string of multiple vehicles over lossy datalinks," in *Decision and Control, 2003. Proceedings. 42nd IEEE Conference on*, vol. 1, pp. 682–687, IEEE, 2003. 2.2

[62] R. Hall and C. Chin, "Vehicle sorting for platoon formation: impacts on highway entry and throughput," *Transportation Research Part C: Emerging Technologies*, vol. 13, no. 5, pp. 405–420, 2005. 2.2

[63] F. Morbidi, P. Colaneri, and T. Stanger, "Decentralized optimal control of a car platoon with guaranteed string stability," in *Control Conference (ECC), 2013 European*, pp. 3494–3499, IEEE, 2013. 2.2

[64] C. Roncoli, M. Papageorgiou, and I. Papamichail, "Optimal control for multi-lane motorways in presence of vehicle automation and communication systems," 2014. 2.2

[65] Y. Zhao, P. Minero, and V. Gupta, "Disturbance propagation in strings of vehicles with limited leader information," in *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pp. 757–762, IEEE, 2012. 2.2

[66] M. Segata and R. Lo Cigno, "Automatic emergency braking: Realistic analysis of car dynamics and network performance," 2013. 2.2

[67] J. J. Haas, "The effects of wireless jamming on vehicle platooning," 2009. 2.2

[68] K. Pelechrinis, C. Koufogiannakis, and S. V. Krishnamurthy, "Gaming the jammer: Is frequency hopping effective?," in *Proc. 7th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'09)*, (Seoul, Korea), June 2009. 2.3.1

[69] A. Molisch, *Wireless Communications*. John Wiley & Sons, Inc., 2005. 2.3.1

[70] M. Raya, I. Aad, J.-P. Hubaux, and A. El Fawal, "DOMINO: Detecting MAC layer greedy behavior in IEEE 802.11 hotspots," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 1691–1705, Dec. 2006. 2.3.1

[71] P. Tague, D. Slater, G. Noubir, and R. Poovendran, "Linear programming models for jamming attacks on network traffic flows," in *Proc. 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt'08)*, (Berlin, Germany), pp. 207–216, Apr. 2008. 2.3.1

[72] L. Rosenberg and D. Gray, "Anti-jamming techniques for multichannel sar imaging," *IEE Proceedings-Radar, Sonar and Navigation*, vol. 153, pp. 234–242, June 2006. 2.3.1

[73] P. Tague, S. Nabar, J. A. Ritcey, and R. Poovendran, "Jamming-aware traffic allocation for multiple-path routing using portfolio selection," *IEEE/ACM Transactions on Networking*. To appear 2010. 2.3.1

[74] J. Becker, J. D. Lohn, and D. Linden, "An in-situ optimized anti-jamming beamformer for mobile signals," in *IEEE Antennas and Propagation Society International Symposium 2012*, pp. 1–2. 2.3.1

[75] W. Xu, W. Trappe, Y. Zhang, and T. Wood, "The feasibility of launching and detecting jamming attacks in wireless networks," in *Proc. ACM 6th International Symposium on Mobile Ad Hoc Networking and Computing*, (Urbana-Champaign, IL, USA), pp. 46–57, May 2005. 2.3.1

[76] H. Liu, Z. Liu, Y. Chen, and W. Xu, "Localizing multiple jamming attackers in wireless networks," in *Proc. of Int'l Conf. on Distributed Computing Systems*, 2011. 2.3.1

[77] W. Xu, T. Wood, W. Trappe, and Y. Zhang, "Channel surfing and spatial retreats: Defenses against wireless denial of service," in *Proc. of the ACM Workshop on Wireless Security*, (Philadelphia, PA), Oct. 2004. 2.3.1

[78] P. Tague, "Improving anti-jamming capability and increasing jamming impact with mobility control," in *6th IEEE International Workshop on Wireless and Sensor Networks Security (WSNS)*, Nov. 2010. 2.3.1

[79] D. J. Thuente and M. Acharya, "Intelligent jamming in wireless networks with applications to 802.11b and other networks," in *MILCOM'06*, (Washington, DC), Oct. 2006. 2.3.1, 6

[80] M. Wilhelm, I. Martinovic, J. Schmitt, and V. Lenders, "Reactive jamming in wireless networks: How realistic is the threat?," in *Proc. 4th ACM Conference on Wireless Network Security*, (Hamburg, Germany), June 2011. 2.3.1

[81] K. Pelechrinis, M. Iliofotou, and S. Krishnamurthy, "Denial of service attacks in wireless

networks: The case of jammers," *IEEE/ACM IEEE Communication Surveys and Tutorials*, vol. 13, pp. 245–257, May 2011. 2.3.1, 6

[82] B. Awerbuch, A. Richa, and C. Scheideler, "A jamming-resistant mac protocol for single-hop wireless networks," in *Proc. of the 27th ACM symposium on Principles of distributed computing*, (Toronto, Canada), 2008. 2.3.1

[83] M. Manshaei, Q. Zhu, T. Alpcan, T. Başar, and J.-P. Hubaux, "Game theory meets network security and privacy," *ACM transaction on Computational Logic*, vol. 5, 2011. 2.3.2

[84] T. Alpcan and T. Başar, *Network Security: A Decision and Game-Theoretic Approach*. Cambridge University Press, 2010. 2.3.2

[85] M. Felegyhazi and J. Hubaux, "Game theory in wireless networks: A tutorial," tech. rep., Technical Report LCA-REPORT-2006-002, EPFL, 2006. 2.3.2

[86] E. Altman, K. Avrachenkov, and A. Garnaev, "Transmission power control game with SINR as objective function," in *Network Control and Optimization*, pp. 112–120, Springer, 2009. 2.3.2

[87] K. Firouzbakht, G. Noubir, and M. Salehi, "On the capacity of rate-adaptive packetized wireless communication links under jamming," in *Proceedings of the fifth ACM conference on Security and Privacy in Wireless and Mobile Networks*, pp. 3–14, ACM, 2012. 2.3.2

[88] M. Li, I. Koutsopoulos, and R. Poovendran, "Optimal jamming attacks and network defense policies in wireless sensor networks," in *IEEE 26th IEEE International Conference on Computer Communications, 2007*, pp. 1307–1315. 2.3.2

[89] R. K. Mallik, R. A. Scholtz, and G. P. Papavassilopoulos, "Analysis of an on-off jamming situation as a dynamic game," *Communications, IEEE Transactions on*, vol. 48, no. 8, pp. 1360–1373, 2000. 2.3.2, 5

[90] E. Altman, K. Avrachenkov, and A. Garnaev, "A jamming game in wireless networks with transmission cost," in *Network Control and Optimization*, pp. 1–12, Springer, 2007. 2.3.2

[91] A. Gupta, A. Nayyar, C. Langbort, and T. Başar, "A dynamic transmitter-jammer game with asymmetric information," in *51st IEEE Annual Conference on Decision and Control, 2012*, pp. 6477–6482. 2.3.2

[92] S. Bhattacharya, A. Khanafer, and T. Başar, "Switching behavior in optimal communication strategies for team jamming games under resource constraints," in *IEEE International Conference on Control Applications 2011*, pp. 1232–1237. 2.3.2

[93] Y. Gwon, S. Dastangoo, C. Fossa, and H. Kung, "Competing mobile network game: Embracing antijamming and jamming strategies with reinforcement learning," in *IEEE Conference on Communications and Network Security*, pp. 28–36, IEEE, 2013. 2.3.2

[94] A. Garnaev and W. Trappe, "An eavesdropping game with SINR as an objective function," in *Security and Privacy in Communication Networks*, pp. 142–162, Springer, 2009. 2.3.2

[95] Z. Han, N. Marina, M. Debbah, and A. Hjorungnes, "Physical layer security game: How to date a girl with her boyfriend on the same table," in *IEEE International Conference on Game Theory for Networks, 2009*, pp. 287–294. 2.3.2

[96] Q. Zhu, W. Saad, Z. Han, H. V. Poor, and T. Başar, "Eavesdropping and jamming in next-generation wireless networks: A game-theoretic approach," in *IEEE Military Communication Conference, 2011*, pp. 119–124. 2.3.2

[97] M. L. Littman, "Friend-or-foe Q-learning in general-sum games," in *ICML*, vol. 1, pp. 322–328, 2001. 2.3.2

[98] J. Hu and M. P. Wellman, "Nash Q-learning for general-sum stochastic games," *The Journal of Machine Learning Research*, vol. 4, pp. 1039–1069, 2003. 2.3.2

[99] X. Wang and T. Sandholm, "Reinforcement learning to play an optimal Nash equilibrium in team Markov games," in *In Proceedings of the Neural Information Processing Systems: Natural and Synthetic (NIPS) conference*, 2002. Extended version at http://www.cs.cmu.edu/ sandholm/oal.ps. 2.3.2

[100] M. Kearns, Y. Mansour, and S. Singh, "Fast planning in stochastic games," in *Proceedings of the Sixteenth conference on Uncertainty in artificial intelligence*, pp. 309–316, Morgan Kaufmann Publishers Inc., 2000. 2.3.2

[101] S. Ganzfried and T. Sandholm, "Computing an approximate jam/fold equilibrium for 3-player no-limit Texas Hold'em tournaments," in *International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2008. 2.3.2

[102] S. Ganzfried and T. Sandholm, "Computing equilibria in multiplayer stochastic games of imperfect information," in *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI)*, 2009. 2.3.2

[103] D. E. Kirk, *Optimal control theory: an introduction*. Courier Corporation, 2012. 3

[104] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. 3, 3.2

[105] J. Zhang, "A survey on trust management for vanets," in *Advanced Information Networking and Applications (AINA), 2011 IEEE International Conference on*, pp. 105–112, IEEE, 2011. 4.3.3

[106] J. Rodgers, "Pulse radar systems," 1962. US Patent 3,029,429. 5

[107] E. Altman, K. Avrachenkov, R. Marquez, and G. Miller, "Zero-sum constrained stochastic games with independent state processes," *Mathematical Methods of Operations Research*, vol. 62, no. 3, pp. 375–386, 2005. 5

[108] E. Altman, K. Avrachenkov, and A. Garnaev, "Jamming in wireless networks: The case of several jammers," in *IEEE International Conference on Game Theory for Networks, 2009*, pp. 585–592. 5.2

[109] Y. E. Sagduyu, R. Berry, and A. Ephremides, "Mac games for distributed wireless network security with incomplete information of selfish and malicious user types," in *IEEE International Conference on Game Theory for Networks, 2009*, pp. 130–139. 5.2

[110] B. DeBruhl and P. Tague, "Keeping up with the jammers: Observe-and-adapt algorithms

for studying mutually adaptive opponents," *Pervasive and Mobile Computing*, 2014. 5.3.3

[111] F. Lewis, S. Jagannathan, and A. Yesildirek, *Neural Network Control of Robot Manipulators and Nonlinear Systems*. Taylor and Francis, 1999. 6.2.3

[112] "IEEE 802.15.4-2006," 2006. `http://standards.ieee.org/getieee802/download/802.15.4-2006.pdf`. 6.3, 6.3.1

[113] B. DeBruhl and P. Tague, "Digital filter design for jamming mitigation in 802.15.4 communication," in *20th IEEE International Conference on Computer Communication Networks (ICCCN'11)*, Aug. 2011. 6.3.2

[114] "Ettus research LLC," 2013. `http://www.ettus.com/`. 6.4

[115] "GNU radio," 2013. `http://gnuradio.org/`. 6.4

[116] T. Schmid, O. Sekkat, and M. Srivastava, "An experimental study of network performance impact of increased latency in software defined radios," in *WiNTECH'07*, (Montreal, Quebec, Canada), Sept. 2007. 6.4

[117] G. Huang, Q. Zhu, and C. Siew, "Extreme learning machine: Theory and applications," *Neurocomputing*, vol. 70, Dec. 2006. 7.1