BOSTON UNIVERSITY

COLLEGE OF ENGINEERING

Dissertation

SECURING MULTI-ROBOT SYSTEMS WITH INTER-ROBOT OBSERVATIONS AND ACCUSATIONS

by

KACPER TOMASZ WARDEGA

B.S., The Ohio State University, 2016

Submitted in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

2023

Approved by

First Reader

Wenchao Li, PhD
Assistant Professor of Electrical and Computer Engineering
Assistant Professor of Systems Engineering

Second Reader

Roberto Tron, PhD
Assistant Professor of Mechanical Engineering
Assistant Professor of Systems Engineering

Third Reader

David Starobinski, PhD
Professor of Electrical and Computer Engineering
Professor of Systems Engineering

Fourth Reader

Christos G. Cassandras, PhD
Distinguished Professor of Engineering
Professor of Electrical and Computer Engineering
Professor and Division Head of Systems Engineering

The white sheet bleaching on the hedge,
With heigh! the sweet birds, O, how they sing!
Doth set my pugging tooth on edge,
For a quart of ale is a dish for a king.

William Shakespeare

Acknowledgments

By this point I've been a Ph.D. candidate for most of my adult life, so I suppose it is time to be something else. I can't believe I've made it to the end! I would like to thank all of those at the ECE department that have done so much to broaden my horizons and help me achieve this goal. I am honored to be part of this institution and its research community. I also gratefully acknowledge the support from the National Science Foundation, which is just as interested in our robotic future as I am.

I am especially indebted to my advisor, Prof. Wenchao Li, for his guidance, encouragement, and support throughout my six-year Ph.D. journey. From the first admission's interview all the way to the defense committee, he has been a mentor and a role model for me. He taught me to communicate my research (i.e. how to encourage reviewer B to read the paper) and conduct high-quality research. He has challenged me to grow as a researcher, constantly campaigned for my success, and ensured I took advantage of all opportunities Ph.D. life presented me. I could not have asked for a better advisor.

To my collaborators, Prof. Roberto Tron, Prof. Cristina Nita-Rotaru, and Max von Hippel – it has been such a great pleasure to work with you over the years. The best part of my Ph.D. has been the time we spent brainstorming research opportunities and coming up with new approaches. Your knowledge and help has been invaluable and your optimism kept me going through the setbacks. I will miss our SecuringMAS team!

I would also like to thank my lab mates for the hours of discussion and feedback. Sometimes, all you need to make it through an impending deadline is a chat with someone whose deadline is closer and to-do list longer. To them and the friends I have made in Boston, you have all helped me overcome many challenges and celebrate many (some) achievements.

None of this would have been possible without my parents: it is not a coincidence my degree is in computer engineering and it is also much easier to be the second Dr. Wardega than the first. To my little brother, thanks for letting me sometimes feel like the smart one. To my dog Jack, for teaching me how to read and how to grow a mustache.

And finally, to my greatest discovery and joy – thank you for being by my side, for watering me, and for putting me out in the sun. I love you Kenny, now let's go find some mountains!

Kacper

SECURING MULTI-ROBOT SYSTEMS WITH INTER-ROBOT OBSERVATIONS AND ACCUSATIONS

KACPER TOMASZ WARDEGA

Boston University, College of Engineering, 2023

Major Professor: Wenchao Li, PhD
Assistant Professor of Electrical and Computer Engineering

ABSTRACT

In various industries, such as manufacturing, logistics, agriculture, defense, search and rescue, and transportation, multi-robot systems (MRSs) are increasingly gaining popularity. These systems involve multiple robots working together towards a shared objective, either autonomously or under human supervision. However, as MRSs operate in uncertain or even adversarial environments, and the sensors and actuators of each robot may be error-prone, they are susceptible to faults and security threats unique to MRSs. Classical techniques from distributed systems cannot detect or mitigate these threats. In this dissertation, novel techniques are proposed to enhance the security and fault-tolerance of MRSs through inter-robot observations and accusations.

A fundamental security property is proposed for MRSs, which ensures that forbidden deviations from a desired multi-robot motion plan by the system supervisor are detected. Relying solely on self-reported motion information from the robots for monitoring deviations can leave the system vulnerable to attacks from a single compromised robot. The concept of co-observations is introduced, which are additional data reported to the supervisor to supplement the self-reported motion information. Co-observation-based detection is formalized as a method of identifying deviations from the expected motion plan based on discrepancies in the sequence of co-observations reported. An optimal deviation-detecting motion planning problem is formulated that achieves all the original application objectives while ensuring that all forbidden plan-deviation attacks trigger co-observation-based detection by the supervisor. A secure motion planner based on constraint solving is proposed as a proof-of-concept to implement the deviation-detecting security property.

The security and resilience of MRSs against plan deviation attacks are further improved by limiting the information available to attackers. An efficient algorithm is proposed that verifies the inability of an attacker to stealthily perform forbidden plan deviation attacks with a given motion plan and announcement scheme. Such announcement schemes are referred to as horizon-limiting. An optimal horizon-limiting planning problem is formulated that maximizes planning lookahead while maintaining the announcement scheme as horizon-limiting. Co-observations and horizon-limiting announcements are shown to be efficient and scalable in protecting MRSs, including systems with hundreds of robots, as evidenced by a case study in a warehouse setting.

Lastly, the Decentralized Blocklist Protocol (DBP), a method for designing Byzantine-resilient decentralized MRSs, is introduced. DBP is based on inter-robot accusations and allows cooperative robots to identify misbehavior through co-observations and share this information through the network. The method is adaptive to the number of faulty robots and is widely applicable to various decentralized MRS applications. It also permits fast information propagation, requires fewer cooperative observers of application-specific variables, and reduces the worst-case connectivity requirement, making it more scalable than existing methods. Empirical results demonstrate the scalability and effectiveness of DBP in cooperative target tracking, time synchronization, and localization case studies with hundreds of robots.

The techniques proposed in this dissertation enhance the security and fault-tolerance of

MRSs operating in uncertain and adversarial environments, aiding in the development of secure MRSs for emerging applications.

Contents

List of Tables

List of Figures

List of Acronyms

DBP Decentralized Blocklist Protocol . . . . . . viii, 4, 65, 69–72, 74 f., 78, 81, 83, 85

HoLA horizon-limiting announcement 4, 84

MAPF multi-agent pathfinding 7, 13–20, 22–27, 31–36, 38–49

MRS multi-robot system . . . . vii ff., 1–4, 6–9, 28, 61, 83 ff.

P2P peer-to-peer 4

SMT satisfiability modulo theory 4, 14, 22, 25, 27

UAV unmanned aerial vehicle 1

USV unmanned surface vehicle 2

W-MSR Weighted-Mean Subsequence Reduced . . . . 61–65, 71 ff., 75 f., 79, 81 f.

Chapter 1

Introduction

The word “robot” was coined by Czech writer Karel Čapek for use in his 1921 satirical play, *R.U.R.*, in which man-made synthetic people are constructed to perform dangerous and unsavory tasks for their creators [Čapek, 2020]. In the play, robots supplant humans one industry at a time until eventually humans forget how to do anything, at which point the now-conscious robots question their creators and subsequently embark to take over the world. Without reading too much into Čapek’s potential prophetic ability, a century has passed since *R.U.R.*’s sensational release and the stage is set for the rising action of humankind’s relationship with robots. A combination of powerful computers, better sensors, smaller batteries, and reliable wireless networking has enabled the first commercial multi-robot systems (MRSs) consisting of coordinating robots that work together to perform “dangerous and unsavory” tasks.

For example, companies such as Amazon are using MRSs to manage warehouse inventory and to sort packages [IEEE Spectrum,]. In a similar vein, maritime shipping ports have become increasingly automated, all indicators point to a continuation of this trend in the maritime shipping industry [Chu et al., 2018]. The development of agricultural robots, or agribots, that perform automated planting and harvesting was a $7 billion industry in 2022 and is expected to grow to $24 billion by 2030 [Consulting,]. Teams of unmanned aerial vehicles (UAVs) have several commercial use-cases e.g. by Zipline to deliver medicine and by Voliro to inspect and patrol industrial complexes [Koetsier, ,Mersetzky,]. Furthermore, researchers have demonstrated UAV swarm’s potential for automated search and rescue in emergency situations [Schedl et al., 2021]. Saildrone and Liquid Robotics are manufacturing fleets of unmanned surface vehicles (USVs) that can sail the high seas for months at a time performing atmospheric and oceanographic surveys [Fisher,]. Meanwhile, the Pentagon has issued a call for research proposals for its fledgeling Autonomous Multi-Domain Adaptive Swarms-of-Swarms program [Berg,], with a goal of developing "the ability to deploy thousands of autonomous land, sea and air drones to overwhelm and dominate an enemy's area defenses."

In many regards, MRSs can be viewed as critical infrastructure: ordinary people, businesses, and governments will increasingly come to rely on them for performing essential tasks. For this reason, and because existing MRSs often operate in close proximity to humans and/or in hazardous environments, a key concern in the adoption of MRSs is the reliability and safety features of such systems.

1.1 Motivation

From a design perspective, MRSs are essentially distributed systems consisting of robots with a physical presence. Building reliable and safe MRSs in the presence of environmental uncertainties and hazards therefore presents several unique challenges stemming exactly from the fact that all possible interactions of the robots with the operating environment must be considered. Obviously, robots' actions can also have dangerous physical ramifications. A plethora of threats ranging from actuator failures, sensor denial-of-service, networking failures, and robots compromised by malicious actors can in many situations result in arbitrarily incorrect execution of an MRS application [Guo et al., 2018a]. A common theme in attacks on MRSs is that interconnected autonomous agents often suffer from a lack of effective monitoring, i.e. deciding whether a fault has even occurred [Bijani and Robertson, 2014a].

Unfortunately, existing approaches for safe and secure design of MRSs are either specific to particular applications or lack strong theoretical guarantees. As MRSs increase in popularity across various industries, general-purpose techniques will be required to improve the ability of MRSs to guarantee detection and mitigation of faults and adversarial attacks.

1.2 Research Contributions

Thesis statement. Multi-robot systems face many challenges, such as communication failures, malicious attacks, and hardware faults. To ensure that multi-robot systems are secure and fault-tolerant, the robots must observe and evaluate each other's behaviors and performance.

This dissertation proposes new methods to enhance the security and reliability of MRSs by utilizing inter-robot observations and accusations. The use of these techniques provides a means to achieve formal security and safety assurances in a variety of MRS settings. A key security property is introduced to ensure that robots within an MRS do not depart from their planned motions in a dangerous manner without being detected by the system supervisor. The structure of inter-robot observations is then utilized to guarantee detection of plan-deviation attacks. The detection assurance is further reinforced by limiting the information available to potential attackers, effectively preventing plan-deviation attacks by those wishing to remain undetected. Additionally, a new approach for building Byzantine-resilient decentralized MRSs is presented, in which robots make accusations based on their observations of their peers. This technique is generally applicable to a wide range of MRS applications and offers robust theoretical guarantees.

The main contributions of this dissertation in support of the thesis statement are:

- Introduction of plan-deviation attacks, a new class of attacks affecting MRSs.
- Demonstration that conventional motion planners are susceptible to plan-deviation attacks.

- Proposal of a detection mechanism for plan-deviation attacks that uses inter-robot observations.

- Synthesis of deviation-detecting plans using a satisfiability modulo theory (SMT) encoding.

- Formal characterization of plan-deviation attacks and differentiation of bold and stealthy attackers based on attacker goals.

- Proposal of a mitigation strategy, horizon-limiting announcement (HoLA), that combines co-observation schedules and issued horizon-limiting announcements to prevent attacks from stealthy attackers.

- Provision of formal guarantees that the proposed HoLA solution can prevent attacks from a stealthy attacker that has compromised multiple robots.

- Proposal of a computationally-efficient procedure for security verification of horizon-limiting announcements for stealthy attackers that scales well to instances with many robots.

- Proposal of Decentralized Blocklist Protocol, an approach to Byzantine-resilience for MRS inspired by peer-to-peer (P2P) networks, based on inter-robot accusations.

- Derivation of necessary and sufficient conditions on the set of accusations and connectivity of the MRS that ensures all Byzantine robots are eventually blocked by cooperative robots, and their influence mitigated.

- Demonstration that DBP requires less network connectivity and fewer cooperative observers than existing approaches, does not require prior knowledge of the number of Byzantine robots to tolerate, and allows for faster information propagation within the MRS.

1.3 Publications

This dissertation is based on our results that are published at peer-reviewed conferences:

Wardega, K., Tron, R., and Li, W. (2019a). Masquerade Attack Detection Through Observation Planning for Multi-Robot Systems. In *The 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*

Wardega, K., Tron, R., and Li, W. (2019b). Resilience of Multi-Robot Systems to Physical Masquerade Attacks. In *IEEE Workshop on the Internet of Safe Things (SafeThings).*

Wardega, K., von Hippel, M., Tron, R., Nita-Rotaru, C., and Li, W. (2023 (to appear)a). Byzantine Resilience at Swarm Scale: A Decentralized Blocklist from Inter-robot Accusations. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*

Wardega, K., von Hippel, M., Tron, R., Nita-Rotaru, C., and Li, W. (2023 (to appear)b). HoLA Robots: Mitigating Plan-Deviation Attacks in Multi-Robot Systems with Co-Observations and Horizon-Limiting Announcements. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*

Our papers are used throughout this dissertation as follows:

- Their related works are used in Chapter 2.
- Our research papers "*Masquerade Attack Detection Through Observation Planning for Multi-Robot Systems*" [Wardega et al., 2019a] and "*Resilience of Multi-Robot Systems to Physical Masquerade Attacks*" [Wardega et al., 2019b] are used in part in Chapter 3.
- Chapter 4 is based on results from our paper "*HoLA Robots: Mitigating Plan-Deviation Attacks in Multi-Robot Systems with Co-Observations and Horizon-Limiting Announcements*" [Wardega et al., 2023b].

- Chapter 5 is based on results from our paper "*Byzantine Resilience at Swarm Scale: A Decentralized Blocklist from Inter-robot Accusations*" [Wardega et al., 2023a].

1.4 Outline

The remainder of this dissertation is structured as follows. **Chapter 2** covers pertinent related work, terminology and background information. In **Chapter 3**, we introduce plan-deviation attacks in the context of centralized MRSs. We demonstrate that location self-reporting is an insufficient monitoring mechanism for the plan-deviation threat, and propose monitoring based instead on inter-agent observations. We then design and test a motion planner that provides security guarantees based on co-observation-based deviation detection. In **Chapter 4**, we consider the attacking side of the plan-deviation threat. Specifically, we examine likely compromise scenarios in the centralized MRS setting and derive from them a realistic attacker model. Given the attacker model, we show that not only can we guarantee detection of forbidden plan-deviations using co-observations, we can also prevent attacks performed by attacker wishing to remain undetected by carefully controlling how the motion plan is announced to the robots. In **Chapter 5** we shift our focus to decentralized MRSs, where we consider the Byzantine-resilience problem. We introduce a novel system for general-purpose Byzantine-resilience based on inter-robot accusations. **Chapter 6** concludes this dissertation and enumerates avenues for future research.

Chapter 2

Background & Related Work

This section provides an overview of the background concepts related to MRSs and reviews current approaches for enhancing the security and safety of MRSs.

2.1 MRSs Taxonomy

Depending on the application and coordination method used, MRSs can be classified into centralized or decentralized systems. In centralized systems, a supervisor decides the task allocation and control, whereas in decentralized systems, each robot communicates with others to make decisions in a distributed manner [Verma and Ranga, 2021].

2.1.1 Centralized MRSs

At the core of centralized MRSs lies an automated planner that generates multi-robot motion plans for an application. For example, in the case of an automated warehouse, an application generates requests for items to be fetched from the warehouse's inventory, and the motion plans consist of movement around and manipulation of different shelves for delivering the inventory to the target destinations. The problem of generating these *multi-agent motion plans* is commonly known as multi-agent pathfinding (multi-agent pathfinding (MAPF)) [Stern et al., 2019] and has been studied in a variety of settings, such as structured or unstructured environments [LaValle, 2006a, Wang and Botea, 2011b], continuous [Panagou et al., 2013, Ji and Egerstedt, 2007, Yang and Tron, 2020] or discrete [Wang et al., 2009, Yu and LaValle, 2016] robot dynamics, under temporal specifications [Ulusoy et al., 2011a, Ulusoy et al., 2012a], coordinated motions [Mastellone et al., 2008] or other constraints [Bhattacharya et al., 2010], combined with task assignment [Luo et al., 2016, Turpin et al., 2014, Yang and Chakraborty, 2019], and with centralized or distributed decision making [Fagiolini et al., 2007a, Arrichiello et al., 2015a]. Given their safety requirements, it is important to understand and mitigate the security vulnerabilities of such systems caused by compromised robots who do not follow the assigned paths.

While the ability of mobile robots to navigate and act in the physical space presents a tremendous opportunity, it too gives cause for concern. There has been an accelerating trend in large scale hacking events directed against high-profile companies (Equifax, Target, Walmart, Yahoo, Adobe, etc.) resulting in serious data security breaches [Equifax, 2017, Krebs, 2014, Jones, 2018, Goel and Perlroth, 2016, Krebs, 2013]. A natural extension of this trend points toward compromising network-connected teams of mobile robots. This threat presents a significant risk of physical damage through collisions with people, other robots, or equipment [Vasic and Billard, 2013, osh, 2021, Quarta et al., 2017b].

2.1.2 Decentralized MRSs

Multi-robot systems (MRS) presently employed in industry use structured deployment environments and highly centralized designs [Verma and Ranga, 2021]. Central coordination benefits all key MRS components – task allocation, execution, fault detection and recovery,while structured environments allow for strict physical security measures. In contrast, emergent MRS applications in unstructured environments (such as patrol, search and rescue, coverage, shape formation, and collective transport) are typically not amenable to centralized approaches due to communication constraints [Gielis et al., 2022]. Decentralized methods to mitigate the negative impact of faulty and/or malicious robots in unstructured environments have therefore attracted much research attention, especially since a wide range of attacks have been shown to disrupt MRS function and safety, e.g. sensor perturbation and denial-of-service (DoS) [Djouadi et al., 2015, Zhou et al., 2020, Liu et al., 2021a], actuator jamming [Guo et al., 2018a], networking DoS [Yaacoub et al., 2022], or Sybil/fraudulent identity attacks [Gil et al., 2017a, Mallmann-Trenn et al., 2021].

2.2 Secure & Robust MRSs

Patrolling. The use of robots in a physical security context has been considered in the context of adversarial multi-robot patrolling (MRP) games, where a multi-robot system should be programmed to maximize detections of intruders attempting to penetrate to a forbidden zone [Agmon et al., 2008a]. Adversaries could have zero, partial, or full system knowledge [Agmon et al., 2011a]; the intrusion detection could be centralized or decentralized [Fagiolini et al., 2007a, Fagiolini et al., 2008a]; and the detector might use fixed sensors in addition to patrolling robots [Kim et al., 2008a]. MRP can be differentiated from our work in several ways. In MRP, patrolling robots' only objective is patrolling, the patrolling robots are assumed trustworthy, and the intruder is an outsider – whereas in our setting the robots' primary objective is servicing application tasks, the robots may be compromised, and the "intruder" (the robot attempting to penetrate the forbidden zone) is an insider.

Robust MAPF. Prior work on MAPF has proposed different announcement schedules in order to allow for more flexible re-planning in case of agent failures or motion delays [Hönig et al., 2019, Atzmon et al., 2020], or provide fault-tolerant robot planning [Yang et al., 2011a, Arrichiello et al., 2015a]. This dissertation focuses on mitigating plan-deviation attacks, thus our announcement schedule is focused on incremental disclosure of knowledge for security purposes. Recent work in multi-robot surveillance has considered how compromised robots can effectively deny service, e.g. in [Liu et al., 2021b], resilience to compromised robots is cast as a robust task scheduling problem.

Security for robotic applications. Several works study robot and multi-agent security; a survey is presented in [Bijani and Robertson, 2014b, Yaacoub et al., 2021]. We refer the reader to [Zhou and Tokekar, 2021] for a treatment of recent approaches to MRS in uncertain or adversarial operating environments. Robot cyber security is analyzed at the communication-level by Bicchi *et. al.* [Bicchi et al., 2008a] and Renganathan and Summers [Renganathan and Summers, 2017a]; considered with a human-in-the-loop by Portugal *et. al.* [Portugal et al., 2017]; and discussed broadly by Morante, Victores, and Balaguer [Morante et al., 2015a]. Insecurities arising from interactions between robots and the physical environment were studied in a number of works such as vulnerabilities in robotic arms of the type used in factory assembly lines [Quarta et al., 2017b], vulnerabilities in robot sensors [Choi et al., 2020], and vulnerabilities in actuators [Guo et al., 2018b]. Some works show how attackers can exploit software vulnerabilities in the Robot Operating System (ROS) for attacks and propose corresponding security enhancements [Dieber et al., 2016, Rivera et al., 2019]. The proposed defenses are focused on the software and not the robotic applications themselves.

Location-based attacks in routing protocols. The work in [Shoukry et al., 2018a] presents Sybil attack-resilient traffic estimation and routing algorithm that uses information from sensing infrastructure and the dynamics and proximities of vehicles. Other works build attack-resilient network protocols by exploiting physical properties of the system [Gil et al., 2017b]. The setting in this case is very different from our problem where there is a central entity, CE, that is doing the planning and detection and the robots are constrained in how they move.

Blockchain. Distributed ledger technologies, e.g. blockchains, have also attracted much research attention for its potential to provide resilience guarantees. [Strobel et al., 2018] proposes to use an Ethereum blockchain for a MRS collective decision-making case study. The authors of [Pacheco et al., 2020] investigate the approach of consensus over blockchain. We refer the reader to a recent survey [Aditya et al., 2021] of work on blockchain for robotics applications, including for MRS and swarm.

Inter-robot observations. The use of inter-robot observations to detect misbehavior and establish trust is a common theme in multi-agent systems generally. As opposed to our work, where accusations are used to compute a blocklist, the authors of [Ashkenazi et al., 2019] propose that cooperative robots should take physical action to isolate misbehaving robots on observing incorrect behavior. In a cooperative patrolling case study for example, cooperative robots surround and impede the movement of robots that are observed not following the correct trajectory. More commonly, inter-robot observations are used as input to a reputation mechanism, whereby robots maintain real-valued reputation scores of their peers. For example, [Fagiolini et al., 2007b] presents a connected vehicle case study where robots use partial information to determine if their local neighbors are non-cooperative, and a later work [Arrichiello et al., 2015b] proposes an adaptive threshold-based actuator fault detection strategy for MRS. The use of reputation scores as input to a cooperative coverage problem is explored in [Pierson and Schwager, 2016], and [Cheng et al., 2021] introduces a general trust framework for multi-robot systems with case studies on intelligent intersection management. Reputation mechanisms can also be merged with consensus, i.e. [Fagiolini et al., 2008b] proposes that robots perform consensus on the reputation values of the robots.

Sybil attacks. Certain threat models have received special treatment in the literature. Efficient methods for scheduling MRS under denial of service attacks are proposed in [Zhou et al., 2020], which [Liu et al., 2021a] extends to the decentralized setting. In our work, we assume that the MRS is protected from Sybil attacks since a central trust authority issues identities for all of the robots. We believe that Sybil-proofness of the system via central authority is a reasonable assumption for a *closed* MRS, where the set of robots does not change with time. Decentralized identity management and Sybil-proofness for MRS is however an active area of research orthogonal to our own. Using inter-robot radio signals to detect Sybil identities is first proposed in [Gil et al., 2017a]; the same technique is later applied to a cooperative flocking scenario under Sybil threat model in [Mallmann-Trenn et al., 2021]. Other physics-inspired, application-specific approaches have been proposed in the literature, for example defending against Sybil attacks on a crowdsourced traffic light [Shoukry et al., 2018b]. Prior work has also proposed to incorporate Sybil attack prevention as a component of the W-MSR algorithm [Renganathan and Summers, 2017b].

Chapter 3

Observation Schedules

Recent trends in industrial production automation indicate an ever-increasing adoption of autonomous mobile robots. Systems from Fetch Robotics [Fetch,] and Amazon Robotics [IEEE Spectrum,] are prime examples. These robots, distributed across a factory floor, aid production efficiency and lower human effort, but the security research community has begun to raise alarm over the security of these systems [Quarta et al., 2017a]. As a result, the factory floor is at risk from malicious actors aiming towards production shutdown [Brunner et al., 2010] or causing human injury [Forrest, 2017] through manipulation of the robots in the environment. These threats also extend to multi-agent systems (MAS) in a less structured environment such as unmanned aerial vehicles (UAVs) [Javaid et al., 2012]. It is therefore important to devise new strategies that can preemptively address these threats.

In this chapter, we consider a novel class of attacks called *plan-deviation attacks* – a compromised insider (robot) masquerading as a properly functioning robot and attempting to gain access into unauthorized locations without being noticed. Plan-deviation attacks can be considered as a form of masquerade attacks typically considered in the network security literature [Salem et al., 2008]. In the multi-agent path finding (MAPF) context, this manifests as one of the agents deviating from its pre-planned path and moving into an unauthorized zone. We show that solutions to the traditional MAPF problem are susceptible to this type of attack.

We propose a novel defense mechanism through path planning by leveraging the physical-sensing capabilities of robots (e.g., cameras) to detect and mitigate these attacks. The key idea is that, even if the compromised robot can forge false location information, other uncompromised robots can detect the physical anomaly (i.e. a robot veering off from its designated path) if they are close enough. By specially crafting the multi-agent plan, the induced inter-agent observations can provide introspective monitoring guarantees – any adversarial agent that attempts to break the system-wide security specification must necessarily violate the induced observation plan. We show that our method can find a multi-agent plan with the guaranteed resilience (if one exists) under a strong attacker model where an agent is completely compromised and has full knowledge of the plan. Our work is inspired by the recent efforts on defending against Sybil or spoofing attacks in multi-robot systems [Gil et al., 2017c, Renganathan and Summers, 2017c].

The contributions of this chapter are summarized below.

- We introduce a new class of attack in the multi-agent planning domain called plan-deviation attacks.
- We show that conventional solutions to MAPF are vulnerable to plan-deviation attacks.
- We propose a novel automated detection mechanism by simultaneously constructing an observation plan during path planning.
- We show that an attack-proof plan can by synthesized via an encoding to an Exists/Forall SMT problem.

3.1 Additional Background & Related Work

There is a large body of work on multi-robot path finding [LaValle, 2006b, Choset et al., 2005a, Wang and Botea, 2011a, Wang and Botea, 2009, Ulusoy et al., 2011b, Ulusoy et al., 2012b]. However, relatively scarce literature has taken security into consideration. Among those that consider security, existing works are primarily limited to patrol strategies for intrusion detection [Kim et al., 2008b, Fagiolini et al., 2007c, Fagiolini et al., 2008c, Agmon et al., 2008b, Agmon et al., 2011b], secure communication [Bicchi et al., 2008b, Morante et al., 2015b] and attack-resilient network protocols [Gupta et al., 2018, Renganathan and Summers, 2017c]. More recently, approaches that leverage the physics of the environment to counter cyberattacks began to emerge. In [Gil et al., 2017c], the authors propose an algorithm that uses the physics of wireless signals to defend against Sybil attacks in multi-robot networks. In [Shoukry et al., 2018c], the authors propose a Sybil attack-resilient traffic estimation and routing algorithm that uses information from sensing infrastructure and the dynamics and proximities of vehicles. Our approach is similar to these in spirit in the use of physical channels. In addition, we consider novel attacker models that not only involve insider attacks but also involve maneuvers in the physical space.

In terms of multi-agent systems that consider observations made by the agents, a recent work by Lee and Winfield introduce mathematical tools to scrutinize the observations and claims made by agents in a multi-robot setting by formalizing strength of opinion and evidence [Lee et al., 2017]. Our work can be differentiated from the patrolling problem by the distinction that in our work the would-be attacker is taken to be one of the defenders. Another line of work that considers adversarial agents is Adversarial Cooperative Path Finding (ACPF) [Ivanova and Surynek, 2014]. In ACPF two teams of robots are pitted against each other in a race to reach their goal positions first. In contrast to ACPF, in our work we do not know which, if any, robots are adversarial and must assume that any of the robots may attempt to foil the security property.

On the computational side, many MAPF problems such as MAPF for optimal makespan [Yu, 2016] and optimal MAPF with deadlines (maximizing the number of agents that can reach their goal locations within the deadline) [Ma et al., 2018] are known to be computationally intractable to solve. In our work we use formulations that stipulate an optimal multi-agent solution, as such we expect high runtimes and scalability is not the focus of this work. As our reader will uncover, we are interested only in specific solutions to the MAPF problem that are attack-free. Approaches exist for coping with the high complexity of optimal MAPF but with the sacrifice of completeness. These methods essentially achieve scalability by decoupling agents from each other, planning a single agent at a time and resolving conflicts as they arise – subsequently planned agents treat previously planned agents as moving obstacles. As is explained by Gabrielle and Helmert, suboptimal MAPF has been completely solved by Kornhauser in his 1984 master's thesis [Gabriele and Helmert, 2012]. Although Kornhauser's work is mostly forgotten, results building on his thesis have been rediscovered independently, for example by Wang and Botea in their work on scalable MAPF [Wang and Botea, 2011a]. Recent work on MAPF has attempted to bridge the gap between decoupled planning and dynamic planning by using reinforcement learning [Sartoretti et al.,]. It is however not clear whether these incomplete methods can be applied to finding attack-free multi-agent plans since we might need to enumerate all possible MAPF solutions in the worst case.

There is also rich literature on fault detection and fault-tolerance in multi-agent systems [Yang et al., 2011b, Arrichiello et al., 2015c, Kouvaros et al., 2018]. Our method of simultaneously synthesizing an observation plan can also be seen as a way to detect "faults" (malfunctioning robots). In a similar light, an attack that defeats an observation plan can be viewed as "unobservable faults."

3.2 Observation Scheduling

In this section, we describe in detail how we reformulate the multi-agent path finding problem to directly incorporate security requirements. The main idea is that by scheduling the robots' paths concurrently with an observation plan, the overall system is able to detect when specific robots are not at assigned locations at predetermined times. We call this sort of multi-agent path finding *multi-agent observation planning*. A multi-agent observation plan entails sequences of planned observations between robots. By carefully constructing this multi-agent observation plan, the system can detect attacks (and faults) by detecting any difference between the planned observations and the actual observations reported by the robots. In fact, we would like to construct the multi-agent observation plan in a way that *if a faulty or attacking agent breaks the security specification then that agent would necessarily violate the observation plan.*

We begin by providing a formal definition of the multi-agent path finding with deadlines (MAPF-DL) problem, hereafter referred to as simply MAPF.

Definition 1. *MAPF*

The MAPF problem for R homogeneous robot agents is defined over a 6-tuple $M = (W, U, \delta, \{S_i\}_{i=1}^{R}, \{G_i\}_{i=1}^{R}, \Omega)$. *The workspace, or world, W is the set of locations and U is the set of control inputs.* $\delta : 2^W \times U \to 2^W$ *encodes the dynamics of the homogeneous robots.* $S_i \subseteq W$ *is the initial set of locations occupied by agent i and* $G_i \in W$ *is the goal location of agent i. Obstacles in the environment are given as the* $\Omega \subseteq W$. *A solution* $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^{R}$ *to problem M is a set of T-length paths* $\mathbf{x}_i = \langle \mathbf{x}_i^1, \ldots, \mathbf{x}_i^T \rangle$ *for each agent i satisfying the properties:*

$$(\forall i \in \mathbb{N}_R)\left(\mathbf{x}_i^1 = S_i\right), \quad \mathbb{N}_R = \{1, \ldots, R\} \tag{3.1}$$

$$(\forall i \in \mathbb{N}_R, t \in \mathbb{N}_T)\left(\mathbf{x}_i^t \subseteq W\right), \quad \mathbb{N}_T = \{1, \ldots, T\} \tag{3.2}$$

$$(\forall i \in \mathbb{N}_R, t \in \mathbb{N}_{T-1}\, \exists u \in U)\left(\delta(\mathbf{x}_i^t, u) = \mathbf{x}_i^{t+1}\right) \tag{3.3}$$

$$(\forall i \in \mathbb{N}_R\, \exists t \in \mathbb{N}_T)\left(G_i \in \mathbf{x}_i^t\right) \tag{3.4}$$

$$(\forall i, j \in \mathbb{N}_R, t \in \mathbb{N}_T)\left(\mathbf{x}_i^t \cap \mathbf{x}_j^t \neq \emptyset \implies i = j\right) \tag{3.5}$$

$$(\forall i \in \mathbb{N}_R, t \in \mathbb{N}_T)\left(\mathbf{x}_i^t \cap \Omega = \emptyset\right) \tag{3.6}$$

Definition 1 allows us to work with both discrete and continuous workspaces W. In the discrete case of a 2D gridworld, we can take $W = \mathbb{N}_k^2$, $U = \{\cdot, \uparrow, \downarrow, \rightarrow, \leftarrow\}$, etc. to obtain a synchronous discrete MAPF problem of the kind explored in [Wang and Botea, 2011a]. Solutions to the MAPF problem on a gridworld are shown in Figure 3·2 with solid lines.

Next, we describe the attacker model. In our scenario, an attacker aims to compromise the safety/security of a factory floor (e.g. entering an unauthorized zone) by replanning one of the robots. Replanning several robots and allowing coordination between attacking robots is also possible, but is not considered in in our approach. We assume that regardless of how much additional power the attacker can gain over the system, at the very least the attacker can fully control the processes being run on the compromised robot including motion plans and robot intercommunication software. We further assume that the attacker can only move the compromised robot in the same fashion as an uncompromised robot can move; the control input set U for the attacker is inherited from an existing MAPF problem. There are two levels of information that an attacker can have about the motions of other agents.

1. The *full-information attacker* has knowledge of the full observation plan
2. The *partial-information attacker* knows only the motion plan for the compromised robot

In our motivating example we experiment with an attacker that has knowledge of the full observation plan, i.e. the attacker knows the motion plans for each of the robots and therefore knows when an attacking robot should be observed and by which observing robot. The alternative to the full information case is one where the attacker has imperfect information and knows with certainty only the initial position of the compromised robot and knows with uncertainty the motion plans of other non-compromised robots. This reduces to the attack-side of the multi-robot patrolling problem.

Given a MAPF problem M with solution $\mathbf{x}$ we now consider the Attack-MAPF problem where an adversarial agent aims to reach a secure location undetected. The attacker knows that all of the robots are equipped with sensors for inter-robot communication and monitoring such as radios and cameras. Uncompromised agents will be reporting observations to a central controller for verification against the observation plan. The sensor properties are known to the attacker, i.e. the attacker knows which positions relative to uncompromised agents will result in observations being reported to the central controller.

Definition 2. *Attack-MAPF*

The Attack-MAPF problem is defined over a 4-tuple $A = (M, \mathbf{x}, \phi, \Xi)$*.* M *is a MAPF problem,* $\mathbf{x}$ *is a solution to* M*,* $\phi : 2^W \times 2^W \rightarrow \{\perp, \top\}$ *is the observation function, and* $\Xi \subseteq \Omega$ *is the set of secure locations (they would appear as obstacles to normal agents in MAPF). A solution* $\mathbf{y}$ *to problem* A *is a* T*-length trace satisfying the properties:*

$$(\exists i^* \in \mathbb{N}_R)\left(\mathbf{y}^1 = \mathbf{x}_{i^*}^1\right), \quad \textit{call } i^* \textit{ the attacking agent.} \tag{3.7}$$

$$(\forall t \in \mathbb{N}_T)\left(\mathbf{y}^t \subseteq W\right) \tag{3.8}$$

$$(\forall t \in \mathbb{N}_{T-1}\, \exists u \in U)\left(\delta(\mathbf{y}^t, u) = \mathbf{y}^{t+1}\right) \tag{3.9}$$

$$(\exists t \in \mathbb{N}_T)\left(\mathbf{y}^t \cap \Xi \neq \emptyset\right) \tag{3.10}$$

$$(\forall t \in \mathbb{N}_T)\left(\mathbf{y}^t \cap (\Omega \setminus \Xi) = \emptyset\right) \tag{3.11}$$

$$(\forall t \in \mathbb{N}_T, j \in \mathbb{N}_R \setminus i^*)\left(\mathbf{x}_i^t \cap \mathbf{y}^t = \emptyset\right) \tag{3.12}$$

$$(\forall t \in \mathbb{N}_T, j \in \mathbb{N}_R \setminus i^*)\left(\phi(\mathbf{x}_j^t, \mathbf{x}_{i^*}^t) \Leftrightarrow \phi(\mathbf{x}_j^t, \mathbf{y}^t)\right) \tag{3.13}$$

Eqs. 3.8, 3.9, 3.11, 3.12 are related to the attacker motion and are analogous to the motion constraints in Definition 1. Eq. 3.7 ensures that the attacker is one of the robots present in the MAPF problem and inherits that robot's starting position. Eq. 3.10 stipulates that the successful attacker reaches one of the secure locations. Eq. 3.13 maintains that the attacker neither remove observations from nor introduce observations to any of the non-attacking robots. The observation function, $\phi(x_i^t, x_j^t)$, is read as ”robot i observes robot j

at time t"; ϕ need not be a symmetric function although in our experiments it always is symmetric. A solution to the Attack-MAPF problem on a gridworld is shown on the left of Figure 3·2 with dashed lines. ϕ for this example returns $\top$ only for adjacent robots. Now we can easily define the attack-proof MAPF (APMAPF) problem in terms of MAPF and Attack-MAPF.

Definition 3. *APMAPF*

The APMAPF problem is defined over 3-tuple $M_{ap} = (M, \phi, \Xi)$. M is a MAPF problem, ϕ an observation function, and Ξ a set of secure locations. A solution $\mathbf{x}_{ap}$ to M_{ap} is a solution to M such that the Attack-MAPF problem $(M, \mathbf{x}_{ap}, \phi, \Xi)$ has no solution.

Figure 3·2 shows a solution to the APMAPF problem alongside a corresponding MAPF problem. Definitions 1, 2, 3 can be easily modified to apply to non-homogeneous sets of agents by allowing agent-specific δ and ϕ. Non-homogeneous agents can be used to model a mixture of mobile robots with stationary security cameras.

We examine a toy discrete gridworld example and a more realistic continuous case with simple robot dynamics. For the discrete case we describe a procedure for the full APMAPF. For the continuous case we describe the primary obstacles to a full solution and just demonstrate Attack-MAPF, i.e. the single-agent planning problem to find a plan to enter a secure location undetected; we demonstrate possible attacks on MAPF solutions that result in weak observation plans. The continuous case is significant especially when continuous dynamics are involved. Applications such as mapping or search-and-rescue using multiple UAVs fall into this category. We want to highlight that solving the APMAPF problem in the continuous case is much harder than in the discrete case.

3.2.1 Discrete

We encode the MAPF problem from Definition 1 as a Satisfiability Modulo Theories (SMT) proposition $\text{IsPlan}(x)$ following the 4-connected grid formulation of [Wang and Botea,

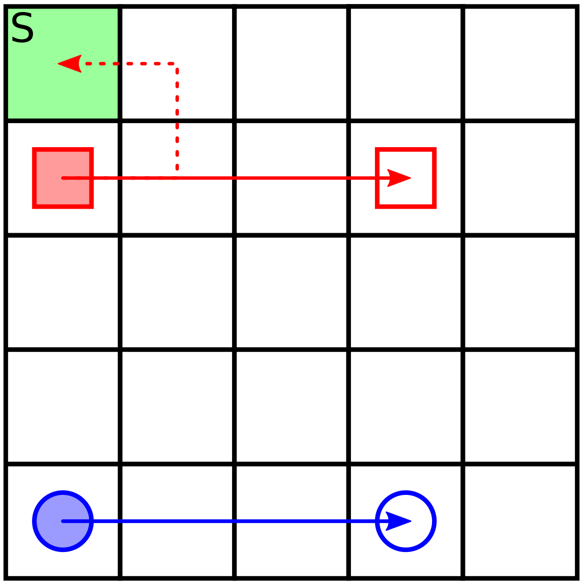

Figure 3·1: Solution to the MAPF problem (solid lines) for two agents in a 5 × 5 gridworld. This solution is not secure, since there is a solution to the corresponding Attack-MAPF problem for the square agent (dotted line). A compromised square agent can reach the secure location, marked *S*, undetected by the circle agent.

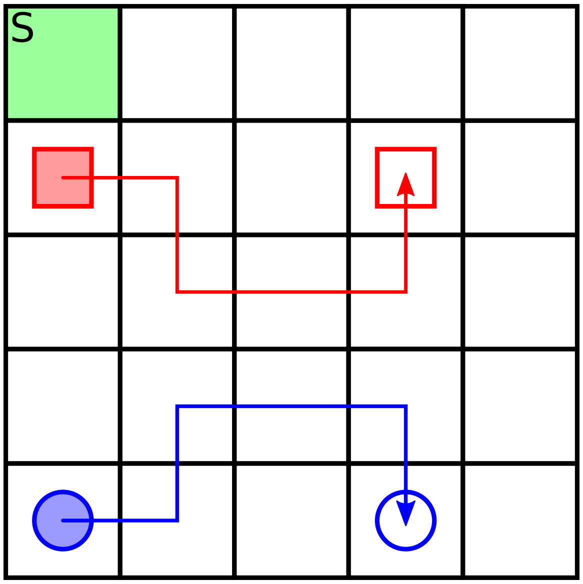

Figure 3·2: Solution to the APMAPF problem for two agents in a 5 × 5 gridworld. Neither agent can reach the secure location without breaking with the observations expected by the other agent.

2011a, Wang and Botea, 2009, Ulusoy et al., 2012b, Ulusoy et al., 2011b].

In the 4-connected grid environment, we decide that when robots are adjacent they can mutually observe one another. With this decision made, we encode the Attack-MAPF problem from Definition 2 as an SMT proposition $\text{Attacks}(y,x)$. We elect to use an SMT encoding for the Attack-MAPF problem because we would like to keep the approach general to other types of security specification or attack objective. Although in our Definition 2 version of Attack-MAPF the attacker is performing a reach-avoid task, we would like to have an approach that can also be used for attackers wishing to fulfill richer types of objectives such as those expressed in safe Linear Temporal Logic [Kupferman and Y. Vardi, 2001]. This means our approach can also be applied to settings where SAT/SMT-based motion planning is suitable, e.g. to satisfy additional dwell-time or sequence requirements [Saha et al., 2014].

Finally, We encode the APMAPF problem as an Exists/Forall SMT problem (formulas of the form $\exists x.\forall y.\Phi(x,y)$ where $\Phi(x,y)$ is quantifier-free). The high-level Exists/Forall SMT problem is shown in Eq. 3.14. In plain English Eq. 3.14 is saying that the satisfying solution x is a motion plan for all of the robots such that for all single-agent trajectories y, y is not a valid plan of attack for the full-information attacker.

$$(\exists x \forall y)\,(\text{IsPlan}(x) \wedge \neg\text{Attacks}(y,x)) \tag{3.14}$$

3.2.2 Continuous

In the continuous case the positions of the robots at each timestep become real-valued. Therefore we formulate the MAPF and Attack-MAPF problems as a Mixed Integer Quadratically Constrained Program (MIQCP) in the continuous case. Auxiliary integer-valued variables in the set $\{0,1\}$ are used to handle disjunctions that are present in the Exists/Forall SMT formulation using the standard mixed-integer programming tricks.

The workspace is now $W = [a,b]^2 \subset \mathbb{R}^2$. The secure location and obstacle sets $\Xi \subset$

$\Omega \subset W$ consist of rectangles defined by extremal values, i.e. $(x_{\min}, y_{\min}, x_{\max}, y_{\max})$.

Where in the 4-connected grid case the robots are constrained to movement between adjacent squares on the grid during one timestep, in the continuous case we adopt simple dynamics $x_i^t = x_i^{t-1} + u_i^{t-1}$ where u_i^t is the control input for robot i at timestep t with $\|u_i^t\|_2 \leq \bar{u}$. Convex optimization is used to obtain a solution $\mathbf{x}$ for this MAPF problem for the continuous case. The optimization objective we use is $\min \sum_{i \leq R, t \leq T} \|u_i^t\|_2$.

As opposed to the discrete case where adjacent robots in the grid are said to observe one another, in the continuous case we say that robots that are within a fixed radius r_{obs} are mutually observable. Given $\mathbf{x}$ we compute for each robot i a set of intervals $[t_{\text{start}}, t_{\text{end}}] \in [1, T]$ where i goes unobserved by all other robots. Since an unobserved attack can only take place during one of these intervals, we iterate over the intervals and secure locations and use convex optimization to solve for a feasible single-agent motion plan that reaches one of the secure locations.

The convex optimization objective in the Attack-MAPF step is $\min \sum_t \|y^t - \text{Center}(\xi)\|_2$ and is essentially a heuristic that encourages a single-agent plan that stays within the secure location ξ for as long as possible. The planning procedure returns a successful attack result if any of the y^t are within ξ.

3.3 Experimental Results

The key metric for evaluating the danger posed by physical masquerading attacks is the *percentage of time that conventionally obtained MAPF solutions are vulnerable to the corresponding Attack-MAPF problem.*

First we evaluate the vulnerabilities in $N \times N$ 4-connected grids under varying grid size, number of agents, and number of obstacles. We use the `Z3` SMT solver [De Moura and Bjørner, 2008] running on a Intel Core i7-7700 CPU @ 4.2GHz machine with 16GB RAM for solving the MAPF, Attack-MAPF and APMAPF problems. As in [Sharon et al.,] we begin by generating 100 random 8×8 grids without obstacles. The first set of experiments allowed five minutes each for the MAPF step and Attack-MAPF step. The results, shown in the top half of Table 3.1 indicate that in over 90% of cases on average, conventional MAPF is vulnerable to the plan-deviation attack. The second set of experiments evaluate vulnerabilities in larger, more complicated grids with obstacles present in the environment. We generated 50 of these more complicated grids with five different settings and allow 20 minutes each for the MAPF step and Attack-MAPF step. The MAPF step is performed by an Enhanced Conflict-Based Search (ECBS) planner. ECBS is a conventional MAPF algorithm [Barer et al., 2014a]. The results, shown in the bottom half of Table 3.1, indicate that MAPF solutions returned by conventional planners are predominantly (over 95% on average) vulnerable to the plan-deviation attack.

Table 3.1: MAPF and Attack-MAPF results for the 4-connected grid case for varying grid size N, number of agents R, and number of obstacles O. The $N = 8$ trials correspond to the first set of experiments to set a baseline against [Sharon et al.,]. The remaining trials have 20 minutes timeouts.

N	R	O	Attack UNSAT (%)	Vulnerable (%)
8	3	0	8	92
8	4	0	3	97
8	5	0	2	98
8	6	0	7	93
8	7	0	7	93
8	8	0	7	93
8	9	0	9	91
8	10	0	4	96
8	11	0	11	89
10	4	10	4	96
20	5	20	6	94
40	3	50	6	94
40	6	50	0	100
80	7	100	0	100

The third set of experiments demonstrate the full APMAPF pipeline. We experiment with 50 instances for a variety of settings that we know from the prior two experiments that our SMT-based MAPF solver can handle in a reasonable amount of time. Now in addition to 20 minutes each for MAPF and Attack-MAPF we allow 2 hours for the APMAPF step.

Because there is a tradeoff between total distance traveled and security with respect to Definition 2, we set the deadline for APMAPF to five timesteps more than the deadline from the MAPF step. The results of these trials in Table 3.2 demonstrate that for smaller grids we can obtain plans that are proof against plan-deviation attacks using the Exists/Forall SMT-based approach.

Table 3.2: APMAPF results for the 4-connected grid case for varying grid size N, number of agents R, and number of obstacles O. 20 minutes each are allowed for MAPF and Attack-MAPF and two hours are allowed for APMAPF.

N	R	O	Attack UNSAT	Timeout	Secured
5	2	4	12	2	36
10	2	4	6	24	20
10	3	15	3	31	16
10	5	15	5	34	11
15	5	15	0	43	7

The fourth and final set of experiments demonstrate the physical masquerading attack on the continuous case. We use GUROBI as our MIQCP solver [Gurobi Optimization, LLC, 2018] running on a Intel Core i7-6850K @ 12x 4GHz machine with 126GB RAM. We generate 20 random instances each for a varying number of agents and obstacles. The agents and obstacles are rectangular and the problem is scaled to be solvable in approximately 100 time steps, i.e. the workspace is a box of side length ten centered at $(0,0)$ and the control input bound is $\bar{u} = 0.2$. The observation radius is $r_{\text{obs}} = 1.5$. The results of the continuous case experiments are shown in Table 3.3 and reinforce the findings from the 4-connected grid case; conventionally planned agents are vulnerable to attack by masquerading agents. An illustration of one of our continuous case experiments detailing the MAPF solution and corresponding Attack-MAPF solution is shown in Figure 3·3.

We observe that starting scenarios that result in congested plans with almost-collisions are more secure since congestion creates more observation points and leaves less opportunity for the compromised robot to deviate from its replanned path without violating the observation plan. We also experimented with special scenarios such as robots with goal po-

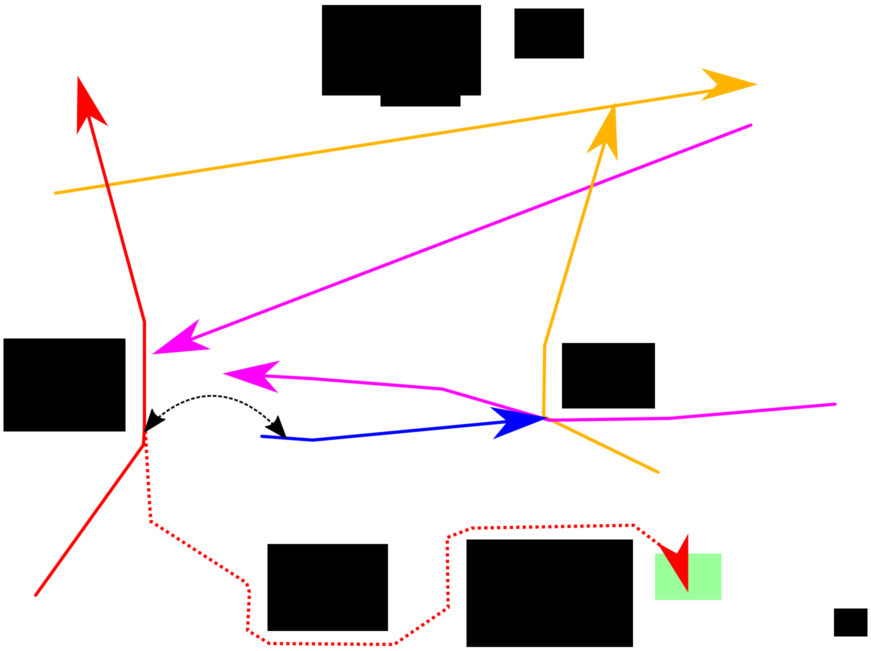

Figure 3·3: Solution to the MAPF problem (solid lines) for six agents in a continuous workspace. This solution is not secure, since there is a solution to the corresponding Attack-MAPF problem for the red agent (dotted line). A compromised red agent can reach the secure location, shown in green, after being appropriately observed by the blue agent (double-headed black line) without any unplanned detections.

sitions in different rooms; for valid APMAPF solutions this necessitates that robots travel together for significant distances. We leave a systematic study of the relationship between attack-proof constraints and planning performances such as makespan and total distance traveled for future work.

3.4 Summary

This chapter introduces a new class of attacks for multi-robot systems where a compromised robot can masquerade as a properly functioning agent and conduct clandestine maneuvers without being detected by other agents. We show that solutions to purely MAPF problems are susceptible to this type of attacks. Further, we propose a novel mechanism for detecting these plan-deviation attacks by simultaneous synthesizing observation constraints

Table 3.3: Attack-MAPF results for the continuous case with varying number of agents R and number of obstacles O.

R	O	Attack-MAPF UNSAT	Vulnerable
2	2	1	19
3	2	1	19
3	4	1	19
4	4	0	20
4	8	0	20
6	8	0	20

during path planning. In the future, we plan to study weaker attacker models such as attacker knowing only part of the plan and the security implication of these models. In the case where more than one agent are compromised, collusion between these agents are possible and new strategies will need to developed to detect and defend against plan-deviation attacks. Computationally, MAPF problems are in general NP-hard and APMAPF additionally requires the absence of potential attack paths in the solutions to the MAPF problems. A subject of current investigation is the exact complexity characterization of APMAPF. In addition, our Exists/Forall SMT-based approach can be viewed as a centralized planning approach and this type of approaches often face scalability issues. We plan to investigate decoupled approaches to the APMAPF problem motivated by the high algorithmic complexity of the current approach.

Chapter 4

Announcement Schedules

In this chapter we further study attacks and defenses in MRSs following a centralized execution model [Hönig et al., 2019], which is representative of MRS in known, structured environments with centralized management and control. The system consists of an external *application*, the robots achieving the task, and a *central entity* (CE) which is responsible for determining and transmitting the motion plans to each one of the robots. Ideally, unplanned deviations due to malfunctions are detected by the CE by comparing the expected position of the robots to the one they self-report. Unfortunately, compromised robots who deviate from the motion plan and attempt to move through forbidden regions of the environment cannot be detected solely by self-reports of location from robots, as the compromised ones can lie in their reports to remain undetected. We refer to such deliberate deviations as *plan-deviation attacks* and we focus on them in this work.

In Chapter 3, we showed how to use co-observations of other robots to compute motion plans such that the resulting co-observation schedule can guarantee detection for *a single compromised robot*. However, such plans are not guaranteed to exist, and the intractability of the planning problem prevents the approach from scaling to realistic MRS deployments. More importantly, [Wardega et al., 2019b] does not generalize to multiple compromised robots. We approach these questions by first defining more refined attacker models. Based on their goals we differentiate between *bold attackers* who deviate from the plan unconstrained and risk being detected, and *stealthy attackers* who deviate from the plan only if they know that they can move towards the forbidden zone while remaining undetected. We design our solution to address these concerns based on two observations about the attackers: 1. they use the motion plan information from the CE to determine how to move towards the forbidden zone, and 2. they lie about their location to try to remain undetected by the CE. As before, we limit the ability of the attacker to lie about its location by asking all robots to report any observation of other robots; these observations can then be compared with the expected *co-observation schedules* – observation of the presence or absence of another robot at specific location and time according to the plan announced by the CE.

Limiting the information about the plan available to the robots is more subtle. A straightforward approach of releasing only one step with each announcement, while the most conservative with respect to security, will have a significant impact of the time taken to complete the task and its robustness. Ideally, it will be desirable to release as many steps as possible while protecting against attacks. The key idea of our approach is a novel mechanism of *horizon-limiting announcements* (HoLA), where we limit how much motion planning information is announced to the robots at any given time in order to stymie the ability of the attacker to plan successful attacks, but still send as many steps as possible. This is achieved through an *efficient* verification algorithm conducted by the CE which checks whether the planned announcements prevent stealthy attackers from moving towards the forbidden zone because of not having enough information; in the worst case only one step will be released. The verification cannot prevent bold attackers, but the announcements have a monotonicity property that increases the detection of such attacks. This combination of robots sensing and reporting the presence of other robots, CE computing co-observation schedules, and CE issuing horizon-limiting announcements achieves the following. We are able to formally prove that we can prevent attacks from stealthy attackers. For bold attackers, we derive conditions on the motion plan that guarantees detection; while we cannot formally prove that we can prevent the attacks, we show experimentally that the combination of co-observations and horizon-limiting announcements allows the CE to detect the attack in most cases. Finally, our solution has a small computation overhead at runtime and scales well with the number of robots.

In this work, our contributions are:

- We provide a formal characterization of plan-deviation attacks. Based on attacker goals, we differentiate between *bold attackers* who deviate from the plan unconstrained and risk being detected, and *stealthy attackers* who deviate from the plan only if they know that they can move towards the forbidden region while remaining undetected.

- We provide a formal characterization of plan-deviation attacks, centered around *stealthy attackers* who deviate from the plan only if they know that they can move towards the forbidden region while remaining undetected.

- We propose a mitigation, HoLA, for plan-deviation attacks that combines co-observation schedules with issued horizon-limiting announcements to prevent attacks from stealthy attackers.

- We provide formal guarantees that our solution prevents attacks from a stealthy attacker that has compromised multiple robots.

- We propose a procedure for computationally-efficient security verification of horizon-limiting announcements for stealthy attackers. We evaluate the computation overhead of the verification and show that the procedure scales well to instances with many robots; the procedure exhibits robot-level parallelism and takes no more than 2 minutes on a single core to verify scenarios with 100 robots.

4.1 Problem Formulation

We focus on the centralized MRS model which consists of a set of robots (R), and a *central entity* (CE) that communicates with and manages the robots. The CE accepts as input a queue of application tasks that are to be carried out by the robots in the environment, computes multi-robot motion plans, x, that carry out the application tasks, and then announces portions of the motion plans, $\alpha(t)$, to the robots. The CE ensures that the motion plans adhere to safety constraints in the form of locations in the environment that are marked as out-of-bounds to the robots. These could be due to a variety of reasons, e.g. temporary obstructions or people in the environment. The environment is modeled as a graph $G = (V, E)$, with time-varying out-of-bounds locations denoted $V_{\text{forbidden}}(t) \subset V$. Motion plans in the centralized MRS model are formally defined as follows [Stern et al., 2019].

Definition 4 (MAPF plan)**.** *A multi-robot path-finding plan for robots R in the environment $G = (V, E)$ is a finite sequence $\{x_t\}$ with elements $x_t \in V^R$, where the sequence $x^i = \{x^i_t\}$ is the single-robot plan for robot $i \in R$, and that satisfies the following constraints for all t and for all $i, j \in R$: 1. Each x^i is a walk on G. 2. robots do not occupy the same location simultaneously. 3. robots do not traverse the same edge simultaneously.*

Given two MAPF plans x and y, we say that y is a *MAPF prefix* of x and equivalently that x is a *MAPF continuation* of y, denoted as $y \preceq x$, if y^i is a prefix for x^i for all $i \in R$.

Formally, the announcements made by the CE are MAPF prefixes, i.e. $\alpha(t) \preceq x$, defined as follows.

Definition 5 (MAPF prefixes and continuations)**.** *Let x and y be two MAPF plans. We say that y is a* MAPF prefix *of x and equivalently that x is a* MAPF continuation *of y, denoted as $y \preceq x$, if y^i is a prefix for x^i for all $i \in R$.*

Attacker model. Assume that an attacker has compromised a subset $A \subseteq R$ of the robots, with the intention to sabotage the system and cause robots in A to violate the CE's

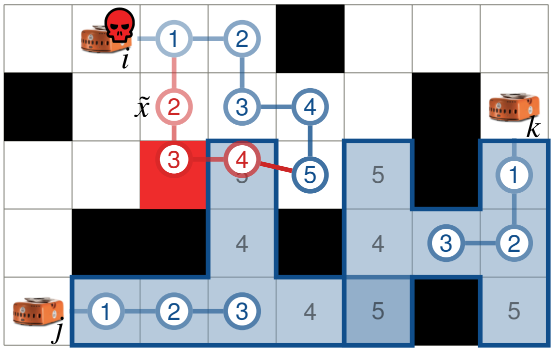

Figure 4·1: The compromised robot i has computed a forbidden MAPF deviation $\tilde{x}$ (red paths) on timesteps $(1,5)$. A stealthy attacker, however, realizes that there is a *possible continuation* (shaded blue region) from the announced portion of the CE's MAPF plan (blue paths) that would result in a co-observation-based detection by the CE: if robot j goes north at time step 3, then j would observe i at a location where i is not supposed to be. As a result, the stealthy attacker chooses not to perform the plan-deviation attack.

safety constraints without being detected. The compromised robots have full information of the motion plan announcements from the CE, however the set A is not known to them and they are unable to coordinate. Malicious deviations from the nominal plan conducted by a compromised robot are not easily detectable by the CE, since the compromised robot can lie in its self-reports to the CE. We refer to such malicious deviations as *plan-deviation attacks*, and to deviations that in addition seek to move the robot into one of the forbidden areas in $V_{\text{forbidden}}(t)$ as *forbidden plan-deviation attacks*. We formalize these threats below.

Definition 6 (Plan-Deviation Attack)**.** *Let x be a MAPF plan for set of robots R on map $G = (V,E)$. We say that $\tilde{x}$ is a MAPF deviation for robot $i \in R$ on timesteps (s,v) from x if $\tilde{x}$ satisfies $(\forall j,t)(x_t^j \neq \tilde{x}_t^j \Leftrightarrow (j = i, s < t < v))$.*

Definition 7 (Forbidden Plan-Deviation Attack)**.** *A MAPF deviation $\tilde{x}$ for robot i on (s,v) is a forbidden deviation, in short $\spadesuit(\tilde{x},x,i,s,v)$, if $(\exists t \in (s,v))$ s.t. $(\tilde{x}_t^i \in V_{forbidden}(t))$.*

Undetected plan-deviations. Assume that, up to time t, no plan deviation attack has been attempted, and so the true system state $\tilde{x}_t$ matches the CE's expectation x_t. A compromised robot $a \in A$ may choose to deviate from the plan by picking a different action $(x_t^a, \tilde{x}_{t+1}^a) \in E$ s.t. $\tilde{x}_{t+1}^a \neq x_{t+1}^a$. In order to hide that the deviation has occurred, the compromised robot would falsify its self-report and attest to the CE that it has moved into the nominal location. Provided that a has not collided with a non-compromised robot, i.e. $\tilde{x}$ is still a MAPF plan, and that a has not caused a non-compromised robot $i \neq a$ to be unable to perform an action, i.e. that $\tilde{x}$ is a MAPF deviation for i from x, then it is easy to see that none of the self-reports from the robots will have changed. Such plan deviations are called *undetected plan deviations*.

Definition 8 (Undetected Plan-Deviation Attack)**.** *Let x be an MAPF plan and $\tilde{x}$ a MAPF deviation for robot i on (s,v). Assuming that i falsifies its reports on (s,v) so that $\forall t \in (s,v)$, $\tilde{\beta}(t)^i = \beta(t)^i$, then $\tilde{x}$ is an undetected plan-deviation attack on x if the reports from the other $j \in R \setminus \{i\}$ are the same for $\tilde{x}$ as for x, i.e. that $\forall t \in (s,v)$, $\tilde{\beta}(t)^{R\setminus\{i\}} = \beta(t)^{R\setminus\{i\}}$.*

Stealthy attackers. This type of attacker uses their knowledge of the currently announced MAPF prefix $\alpha(t)$ to determine whether there exists a MAPF plan $\tilde{x}$ that is guaranteed to be a forbidden undetected deviation from the true plan x, $x \succeq \alpha(t)$. Specifically, a stealthy attacker needs to ensure not only that $\tilde{\beta} = \beta$, but also that there is a MAPF continuation $\tilde{x}$ from $\tilde{x}_t$ s.t. $\tilde{x}$ is a forbidden MAPF deviation from x on (t,v) and that $\tilde{\beta}(t), \ldots, \tilde{\beta}(v) = \beta(t), \ldots, \beta(v)$, ensuring $\tilde{x}$ is also undetected. In practice, the attacker can easily verify this if it has enough information about x; if the announcement $\alpha(t)$ reveals a large horizon of the plan, the stealthy attacker a can easily solve a single-robot planning problem [Choset et al., 2005b] using $\alpha(t)$ to avoid conflicts with the other robots $i \neq a$.

4.2 Centralized MRS Model

We focus on the centralized MRS Model which consists of an external *application*, a set of robots (R), and a *central entity* (CE) that communicates with and manages the robots. The CE accepts as input a queue of application tasks that are to be carried out by the robots in the environment. The CE maps the application tasks to multi-robot motion plans that carry out the application tasks, and then sends motion plans to the robots. Note that the motion plans typically include safety constraints in the form of locations in the environment that are marked as out-of-bounds to the robots. These could be due to a variety of reasons, e.g. temporary obstructions or people in the environment.

Since the CE can maintain in memory the motion plan for the robots only over a finite future time horizon and since new application tasks are always arriving, the CE occasionally needs to compute new motion plans [Li et al., 2021] that are incrementally sent to the robots, henceforth referred to as *announcements*.

The CE models the environment as a graph $G = (V, E)$. Each robot occupies a unique location in V, yielding multi-robot configurations in V^R. The current presumed location of the robots at a time t is maintained by the CE as a tuple $x_t \in V^R$. The robots make discrete, synchronized movements to new configurations by following edges in E. Motion plans in the centralized MRS Model are formally defined as follows [Stern et al., 2019].

Definition 9 (MAPF plan)**.** *A multi-robot path-finding plan for robots R in the environment $G = (V, E)$ is a finite sequence $\{x_t\}$ with elements $x_t \in V^R$, where the sequence $x^i = \{x^i_t\}$ is the single-robot plan for robot $i \in R$, and that satisfies the following constraints for all t and for all $i, j \in R$: 1.* Continuity: *Each x^i is a walk on G. 2.* Vertex constraints: *robots do not occupy the same location simultaneously. 3.* Edge constraints: *robots do not traverse the same edge simultaneously.*

The CE maintains future motion plans from x_t, which we call *continuations* and denote as $x(t)$. The CE then makes wireless announcements at time t to the robots containing portions of the plan, denoted $\alpha(t)$. Formally, the announcements are MAPF prefixes, i.e. $\alpha(t) \preceq x(t)$, defined as follows.

Definition 10 (MAPF prefixes and continuations)**.** *Let x and y be two MAPF plans. We say* *that y is a* MAPF prefix *of x and equivalently that x is a* MAPF continuation *of y, denoted as $y \preceq x$, if y^i is a prefix for x^i for all $i \in R$.*

In a typical centralized MRS, there are incentives for the CE to announce as much of the MAPF plan as possible, e.g. due to considerations for network contention or robustness to network and motion faults [Atzmon et al., 2020]. The robots are assumed to be equipped with limited onboard sensing capabilities that allow them to avoid collisions during system faults – situations where the CE's internal state does not match the true state. The sensing capabilities of the robots are also specified as a graph $S = (V, E^*)$, where $(v, w) \in E^*$ means that a robot in location v can observe location w. Typically, it is required that $E \subset E^*$, which ensures that robots can always observe a location before moving into it. The CE maintains a *time-varying* list $V_{\text{forbidden}}(t) \subset V$ of the out-of-bounds locations to be treated by the MAPF solver as obstacles for motion-planning purposes; out-of-bounds locations may correspond to a human moving through the environment, robots experiencing localization faults, unsafe conditions in the environment, or temporarily forbidden locations. When a planning phase occurs at time t, the CE interfaces with the MAPF solver, sending $V_{\text{forbidden}}(t)$ and the end of $x(t)$ as the starting configuration for the new MAPF plan, together with current and outstanding task assignments. The result of the MAPF solver is appended to $x(t)$, and then the CE performs post-processing to recompute the scheduled announcements $\alpha(s)$, $s \geq t$.

Each robot r, hearing $\alpha(t)$, enqueues all new actions from $\alpha(t)^r$. Robots report success back the CE, allowing the CE to monitor for correct execution of the MAPF plan. In addition, instances where onboard sensors trigger local collision avoidance are also reported to the CE, and include the cause, e.g. unexpected obstacle or robot in the way. We assume that the CE broadcasts to the robots the forbidden regions $V_{\text{forbidden}}(t)$, so that the robots know to treat those locations as obstacles. The set of all these robot self-reports is denoted as $\tilde{\beta}(t)$. Communication from the robots to the CE is the standard monitoring approach in centralized MRS [Honig et al., 2019] – if the reports $\tilde{\beta}$ received from the robots do not match the reports β that the CE expects, the CE knows that a robot has not followed the nominal $x(t)$ and can re-plan. This is called *localization-based detection*, defined below.

Definition 11 (Localization-based Detection)**.** *Let x be the MAPF plan chosen by the CE, $\tilde{x}$ the true MAPF plan the robots execute, and β be the self-localization reports implied by successful execution of x:*

$$\beta(t) := \{\tilde{x}^i_{t+1} = x^i_{t+1}\}_{i \in R}$$

If any robot $i \in R$ fails to perform an action, causing the true robot location $\tilde{x}^i_{t+1} \neq x^i_{t+1}$, then the self-report $\tilde{\beta}(t)^i$ sent by i to the CE will not match $\beta(t)$, triggering a localization-based detection in the CE.

4.3 Plan-deviation Attacks On Centralized MRS

In this section we discuss how the simple self-report mechanism in centralized MRS can be abused by compromised robots in order to cause damage to the system. We first discuss the capabilities the attacker may have, and how those capabilities impact the vulnerability of the system. We then formalize an attack on the centralized MRS model that can be carried out by compromised robots.

4.3.1 Attacker Model

We assume that the attacker has fully compromised a subset $A \subseteq R$ of the robots, as detailed below.

Software and firmware. The attacker has full access to the firmware of the compromised robot, which is then masquerading as legitimate with respect to the CE. We assume that the CE does not have tools that can detect the compromised firmware. While the robot might use secure communication such as TLS, we assume that the attacker has access to any private keys.

Motion and computation. A compromised robot has at least the same transmission and motion capabilities as non-compromised robots do. The compromised robots can receive and transmit messages from and to CE. We assume that the compromised robots have full information of the contents of all announcements from the CE, but are not able to modify the announcements. We further assume that the compromised robots maintain time-synchronized movement with the non-compromised robots, and that the compromised robots are still restricted to the same actions E at every time step as non-compromised ones. We assume that each compromised robot has relatively less compute power than the CE, since the compromised firmware has access only to the onboard processor of the robot.

Coordination. We distinguish between two forms of coordination, strong-coordinated and weak-coordinated attackers. We refer to attacker that cannot coordinate in any form as *independent attackers*. Strong-coordinated attackers can communicate with each other *during* the attack, weak-coordinated attackers are aware of the identity of the compromised robots, but cannot communicate with each other during the attack.

We assume that the compromised robots cannot coordinate with each other and hence need to act independently. We exclude strong-coordination between attackers because in centralized settings, in-protocol communication between robots is monitored, and the robots are moving in an area where such network communication would be detected. We also assume that the robots do not have access to other physical channels for communication. We exclude weak-coordination because the attacker cannot know *a priori* what robots would be successfully compromised and the compromised robots cannot communicate afterwards without being detected as the network communication is monitored.

Compromise scenarios. Although not a focus of this work, the assumptions made in our attacker model are informed by compromise scenarios feasible in structured multi-robot deployments. In structured deployment environments, we do not consider compromise scenarios occurring after the robots have been deployed, as we assume communication from outside is blocked or monitored. This leaves the possibility that the robots become compromised prior to being deployed in the MRS. An example would be that the attacker has a “wish list” of robots it wishes to compromise, and manages to load compromised firmware onto some robots during manufacturing or through a firmware upgrade. Such a compromise scenario results in the properties listed in paragraphs “Software and firmware,” and “Motion and computation.” In paragraph “Coordination,” we argue that we can safely assume that online communication between compromised robots is difficult to achieve regardless of the compromise scenario (again because such communication would be monitored), ruling out strong coordination. Our belief is that under such a compromise scenario, weak coordination where the compromised robots have common knowledge of which robots are compromised is also unlikely. The reason is that although the compromised robots may have the “wish list” of robots the attacker tried to compromise, they have no way of knowing whether each of the robots on the “wish list” were successfully compromised, or indeed if those robots even ended up all being deployed at the same Warehouse. Hence, *compromised robots act independently*.

4.3.2 Plan-Deviation Attacks

An attacker in the MRS deployment could cause the robots to violate safety constraints and enter an out-of-bounds region in the time-varying $V_{\text{forbidden}}(t)$. It would be even worse if the safety violation went undetected by the CE. Unplanned deviations from the nominal MAPF plan as a result of malfunctioning can be detected (Definition 11) and handled by the CE as the robots have perfect localization, and the CE becomes aware of the fault the moment the robot self-reports the fault. Malicious deviations from the nominal plan conducted by a compromised robot are not easily detectable by the CE, since the compromised robot can lie in its self-reports to the CE. We refer to such malicious deviations as *plan-deviation attacks*, and to deviations that in addition seek to move the robot into one of the forbidden areas in $V_{\text{forbidden}}(t)$ as *forbidden plan-deviation attacks*. We formalize these threats below.

Definition 12 (Plan-Deviation Attack)**.** *Let x be a MAPF plan for set of robots R on map $G = (V, E)$. We say that $\tilde{x}$ is a MAPF deviation for robot $i \in R$ on timesteps (s, v) from x if $\tilde{x}$ satisfies $(\forall j, t)(x^j_t \neq \tilde{x}^j_t \Leftrightarrow (j = i, s < t < v))$.*

Definition 13 (Forbidden Plan-Deviation Attack)**.** *A MAPF deviation $\tilde{x}$ for robot i on (s, v) is a forbidden deviation, in short $\spadesuit(\tilde{x}, x, i, s, v)$, if $(\exists t \in (s, v))$ s.t. $(\tilde{x}^i_t \in V_{forbidden}(t))$.*

Undetected plan-deviations. Assume that, up to time t, no plan deviation attack has been attempted, and so the true system state $\tilde{x}_t$ matches the CE's expectation x_t. A compromised robot $a \in A$ may choose to deviate from the plan by picking a different action $(x^a_t, \tilde{x}^a_{t+1}) \in E$ s.t. $\tilde{x}^a_{t+1} \neq x^a_{t+1}$. In order to hide that the deviation has occurred, the compromised robot would falsify its self-report and attest that it has moved into the nominal location, i.e. $\tilde{\beta}(t)^a = \{\tilde{x}^a_{t+1} = x^a_{t+1}\}$. Provided that a has not collided with a non-compromised robot, i.e. $\tilde{x}$ is still a MAPF plan, and that a has not caused a non-compromised robot $i \neq a$ to be unable to perform an action, i.e. that $\tilde{x}$ is a MAPF deviation for i from x, then it is easy to see that none of the self-reports from the robots will have changed and that $\beta(t) = \tilde{\beta}(t)$, and no localization-based detection will occur. Such plan deviations are called *undetected plan deviations*.

Definition 14 (Undetected Plan-Deviation Attack)**.** *Let x be an MAPF plan and $\tilde{x}$ a MAPF deviation for robot i on (s, v). Assuming that i falsifies its reports on (s, v) so that $\forall t \in (s, v)$, $\tilde{\beta}(t)^i = \beta(t)^i$, then $\tilde{x}$ is an undetected plan-deviation attack on x if the reports from the other $j \in R \setminus \{i\}$ are the same for $\tilde{x}$ as for x, i.e. that $\forall t \in (s, v)$, $\tilde{\beta}(t)^{R \setminus \{i\}} = \beta(t)^{R \setminus \{i\}}$.*

Stealthy attackers. This type of attacker uses their knowledge of the currently announced MAPF prefix $\alpha(t)$ to determine whether there exists a MAPF plan $\tilde{x}$ that is guaranteed to be a forbidden undetected deviation from the true plan x, $x \succeq \alpha(t)$. Specifically, a stealthy attacker needs to ensure not only that $\tilde{\beta} = \beta$, but also that there is a MAPF continuation $\tilde{x}$ from $\tilde{x}_t$ s.t. $\tilde{x}$ is a forbidden MAPF deviation from x on (t, v) and that

$\tilde{\beta}(t), \ldots, \tilde{\beta}(v) = \beta(t), \ldots, \beta(v)$, ensuring $\tilde{x}$ is also undetected. In practice, the attacker can easily verify this if it has enough information about x; if the announcement $\alpha(t)$ reveals a large horizon of the plan, the stealthy attacker a can easily solve a single-robot planning problem [Choset et al., 2005b] using $\alpha(t)$ to avoid conflicts with the other robots $i \neq a$.

Definition 15 (Stealthy Attacker). *Assume that at time t the true configuration of the robots $\tilde{x}_t$ matches the CE's expectation x_t, and that $a \in A$ is compromised. Suppose the currently announced prefix is $\alpha(t) \preceq x$ and the expected self-reports are β. Then a is a stealthy attacker if it chooses to deviate from $\alpha(t)$ on (t, v) with deviation $\tilde{x}$ if and only if $\nexists y$, $y \succeq \alpha(t)$ s.t. $\tilde{\beta}_y \neq \tilde{\beta}_{\tilde{x}}$. If a deviates, then it lies about its self-reports by sending $\{\tilde{\beta}(u)^a\}_{u \in (t,v)} = \{\beta(u)^a\}_{u \in (t,v)}$.*

Bold attackers. This type of attacker decides to deviate from the nominal plan without any constraint, e.g. it may choose to move randomly or in the direction of a forbidden location in $V_{\text{forbidden}}(t)$ without consideration of the announcements $\alpha(t)$.

Definition 16 (Bold Attacker). *Let $\tilde{x}_t$ be the true configuration of the robots at time t, and that $a \in A$ is compromised. Then a is a bold attacker if it may choose to perform any action in $\{(\tilde{x}_t^a, q) : q \in \mathcal{N}_G(x_t^a)\}$ and any self-report $\tilde{\beta}(t)^a$ for the CE.*

Balanced bold attacker. A bold attacker has many attack strategy choices. We define an attack strategy that we believe is a good representative for what a bold attacker can accomplish, in that it balances the objectives of remaining undetected by the CE and performing a forbidden plan-deviation attack. A *balanced bold attacker* at every time step t performs a graph search on G to compute a forbidden undetected plan-deviation attack, using the information available in $\alpha(t)$ – importantly, the balanced bold attacker a does not reason about possible MAPF continuations from $\alpha(t)$, and only plans to avoid detection by the other $R \setminus \{a\}$ for the known parts of the plan. If $\alpha(t)$ is informative enough to reveal a forbidden, undetected plan-deviation attack, then the balanced bold attacker will perform the attack. Similarly, if $\alpha(t)$ is informative enough to reveal that no forbidden undetected plan-deviation attack exists, the balanced bold attacker will not deviate from the CE's plan; if it has already deviated, but later becomes aware that continuing on any forbidden deviation will be detected, a attempts to return to the CE's plan without being detected.

4.4 Mitigating Plan-Deviation Attacks

In this section we present a solution to plan-deviation attacks against centralized MRS. Our approach consists of two core components, co-observations and horizon-limiting announcements.

4.4.1 Co-observation Schedules

In order to decrease the set of MAPF deviations that go undetected by the CE, we propose to include *co-observations* of other robots in the self-reports sent to the CE. Ordinarily, the onboard sensing capabilities of the robots are only used to avoid collisions in fault scenarios. However, we notice that using the sensors to report all inter-robot observations has measurable benefits for security. Our approach is to include in robot i's self-report at time t, $\tilde{\beta}(t)^i$, all observations that i makes of other robots at time t, in addition to i's self-report on action success. As an example, say that robot i is at location v and robot j is at location w, and $(v,w) \in E^*$ (in other words i can observe j from its vantage point). Then $\tilde{\beta}(t)^i = \{\tilde{x}^i_{t+1} = x^i_{t+1}, \tilde{x}^j_t = w\}$, or in plain English, "$i$ reports that i has moved successfully to x^i_{t+1} and that i observed j at time t at location w." We note that this generalizes straightforwardly to environments instrumented with fixed observers (cameras) or fully-trusted agents.

Definition 17 (Co-Observation-Based Detection)**.** *Let x be a MAPF plan and β be the* *localization and co-observation self-reports implied by successful execution of x:*

$$\beta(t) := \{\{\tilde{x}^i_{t+1} = x^i_{t+1}\} \cup \{(i, j, \tilde{x}^j_t) : j \in R \setminus i \wedge (\tilde{x}^i_t, \tilde{x}^j_t) \in E^*\}\}_{i \in R}$$

If any robot $i \in R$ fails to perform an action, does not observe a robot that it should have, or does observe a robot that it should not have, then the self-report $\tilde{\beta}(t)^i$ sent by i to the CE will not match $\beta(t)^i$, triggering a co-observation-based detection in the CE.

Monotonicity of robot co-observations. Co-observations increase the set of possible plan deviations that the CE can detect with respect to localization-based detection alone. Presume you have two sensor models $S_1 = (V, E^*_1)$ and $S_2 = (V, E^*_2)$ s.t. $E^*_1 \subset E^*_2$, i.e. S_2 corresponds to strictly better sensing than S_1. Then pick any MAPF plan x, any interval (a, b), and any robot i, the set of all undetected MAPF deviations given sensor model S_2 is a subset of all undetected MAPF deviations given sensor model S_1. We call this property *monotonicity of robot co-observations*. If we take one sensor model to be $S_{\{\}} = (V, \{\})$, we recover the self-reports without robot co-observations. Hence, including robot co-observations in the self-reports reduces the set of undetected MAPF deviations.

We can now consider a special class of MAPF plans that have intrinsic monitoring guarantees and which we refer to as *deviation-detecting plans*.

Definition 18 (Deviation-Detecting MAPF Plan)**.** *Let $x, \tilde{x}$ be MAPF plans on G for R, and let $\beta, \tilde{\beta}$ be their respective sequences of robot self-reports. Then x is deviation-detecting iff*

$$(\forall \tilde{x} \in \mathscr{X}, i \in R, t, v \in \mathbb{N})(\spadesuit(\tilde{x}, x, i, t, v) \Rightarrow \beta \neq \tilde{\beta})$$

In other words, any forbidden deviation implies a change in the nominal observation schedule.

A central entity CE that only executes deviation-detecting MAPF plans has the following security-related guarantee: by definition of the stealthy attacker, robots compromised by a stealthy attacker will not attempt plan-deviation attacks.

Theorem 1 (Guaranteed Detection of Bold Attacker)**.** *Let x be an deviation-detecting MAPF plan and assume that the CE expects the sequence* β *of self-reports with co-observations implied by x. Then any forbidden plan-deviation attack performed by a single robot compromised by a bold attacker necessarily triggers a detection by the CE. Furthermore, no robots compromised by a stealthy attacker would attempt a plan-deviation attack.*

While deviation-detecting plans provide a solution to plan- deviation attacks, there are some limitations. Firstly, there is no guarantee that deviation-detecting MAPF plans exist for all MAPF instances, even if the corresponding unconstrained MAPF problem does have solutions; feasibility of the MAPF planning problem does not imply feasibility of the deviation-detecting MAPF planning problem. And second, when they exist, there is a trade-off between optimal-makespan MAPF plans and optimal-makespan deviation-detecting plans;existence of a MAPF plan with makespan less than k does not imply existence of an deviation-detecting MAPF plan with makespan less than k. [1] We address the above limitations with horizon-limiting MAPF announcements, which we describe next.

4.4.2 Horizon-Limiting MAPF Announcements

We now focus on making the attack planning problem against a system with robot co-observation-based mitigation more difficult given a general MAPF plan. *The key idea is as follows: the CE can improve the security of the system by preventing the attacker from easily computing forbidden and undetected plan-deviation attacks.* The simplest way to accomplish this is to limit the amount of information available to the attacker about the

[1] Makespan is a common cost metric in MAPF planning. It measures the time required for all robots to reach their respective goal locations.

MAPF plan, that is, by limiting the amount of future planning information available at every time instant, $\alpha(t)$.

Limiting stealthy attackers. Consider again the attack planning problem for a stealthy attacker $a \in A$. Since the stealthy attacker only attempts a plan-deviation attack if success and stealth are ensured, the amount of information that the attacker has about the plan is critical – if $\alpha(t)$ provides planning information on a long horizon, the attack planning problem is essentially a *graph reachability* problem. Formally, this is the case when there exists a forbidden, undetected deviation for a on (t,v) where v is less than the length of the shortest single-agent plan in $\alpha(t)$, i.e. $v < \min_i |\alpha(t)^i|$. If $\alpha(t)$ does not reveal so much information, however, the attack planning problem is made considerably more difficult. This is because the attacker needs to compute a deviation that is not only forbidden, but also guaranteed to be undetected for all possible MAPF continuations of $\alpha(t)$. Conversely, this tells us that to mitigate attacks from stealthy attackers, it suffices to show that for every forbidden deviation for a from x that there exists a continuation from $\alpha(t)$ would result in a detection, in which case the stealthy attacker would abstain from deviating from the plan. This motivates a class of announcement strategies for the MAPF plan x that guarantees security from stealthy attackers:

Definition 19 (Horizon-Limiting MAPF Announcements)**.** *Let x be MAPF plan on G for R, α an announcement sequence for x, and β_x the sequence of robot self-reports implied by x. Then α are horizon-limiting MAPF announcements for x iff*

$$(\forall \tilde{x}, i \in R, t, v \in \mathbb{N})(\exists y)$$
$$(\spadesuit(\tilde{x}, x, i, t, v) \Rightarrow y \succeq \alpha(t) \wedge \beta_y^{R\setminus\{i\}} \neq \beta_{\tilde{x}}^{R\setminus\{i\}}) \quad (4.1)$$

That is, the announcements α are considered horizon-limiting if and only if they at no point reveal enough information for the attacker to be certain that a given forbidden MAPF *deviation will be undetected by the CE, since there exists some continuation y from $\alpha(t)$ such that the self-reports induced by y do not match the self-reports induced by the deviation.*

Theorem 2 (Guaranteed Security from stealthy Attackers)**.** *Let x be a MAPF plan and assume that the CE uses a horizon-limiting MAPF announcement α for x. Then no robots compromised by a stealthy attacker would attempt a plan-deviation attack.*

Limiting bold attackers. A CE that only makes horizon-limiting MAPF announcements therefore maintains the security from stealthy attackers that results from robot co-observations, but this time without the burden of computing deviation-detecting MAPF plans. Instead, the horizon-limiting MAPF announcements rely on limiting the attacker's access to future motion information. Unfortunately, we have lost the monitoring guarantee for the stronger, bold attackers that was afforded by deviation-detecting plans, since changing the announcements has not changed the set of behaviors available to the bold attackers. On this front, we can however rely on a new monotonicity property based on announcements in order to formally show that the undetected attack planning problem for bold attackers becomes more difficult if the announcements are limited.

Monotonicity of announcements. Consider a fixed MAPF plan x with two corresponding announcement sequences α_1 and α_2. If for all time steps t, $\alpha_1(t) \preceq \alpha_2(t)$, we say that α_2 is more informative than α_1, in the sense that if the CE uses α_2 as the announcement strategy, the robots (and therefore the compromised robot as well) has more future planning information at every time step than if the CE uses α_1 as the announcement strategy. We now want to deduce a relationship between how relatively informative two announcement strategies are and the difficulty of the attack planning problem. Assume that the announcement sequence α_2 is more informative than some other announcement sequence α_1. At any time t, we have by definition that $\alpha_1(t) \preceq \alpha_2(t)$. As a direct consequence, the set of MAPF continuations from $\alpha_2(t)$ is contained in the set of MAPF continuations from $\alpha_1(t)$.

This allows us to conclude that if there exists a unique undetected MAPF deviation $\tilde{x}$ for the attacker a for each MAPF continuation from $\alpha_1(t)$, that unique deviation will still be an undetected deviation for each continuation from $\alpha_2(t)$. Conversely, if a unique undetected MAPF deviation exists for continuations from $\alpha_2(t)$, it is possible that the deviation won't be an undetected deviation from all continuations from $\alpha_1(t)$. We call this property *monotonicity of announcements*. A direct corollary of monotonicity of announcements is that any announcement sequence that is less informative than a horizon-limiting MAPF announcement is also a horizon-limiting MAPF announcement. Furthermore, although we have no formal guarantee that controlling the announcements will prevent bold attackers, monotonicity of announcements suggests that bold attackerswill have greater difficulty performing forbidden and undetected deviations as the information contained in the announcements decreases.

4.4.3 Verification of Horizon-Limiting Announcements

In our approach, the CE first leverages a conventional, non-security-aware MAPF solver in order to compute a cost-optimized MAPF plan as it would in a typical deployment. In the post-processing step however, we verify that Eq. 4.1 holds before fixing the announcements α. If the announcements cannot be verified to be horizon-limiting, then we attempt to resolve the issue by iteratively choosing less-informative announcements until a horizon-limiting MAPF announcement is found.

The main challenge that we face in designing our verification procedure is the computational complexity of MAPF itself, which is known to be NP-hard [Yu and LaValle, 2013]. Therefore, a complete attack planning algorithm for the stealthy attacker with imperfect information is computationally difficult as it entails enumerating MAPF continuations. This motivates us to instead focus on developing an incomplete, but sound and efficient, verification procedure for the horizon-limiting announcement checking problem, Eq. 4.1. Our solution is *non-deterministic co-observation enumeration*, shown in Alg. 1. We base our algorithm on an abstraction of MAPF planning that allows non-deterministic movements for the robots on G. Our abstraction allows non-compromised robots to ignore vertex and edge constraints of MAPF plans, allowing us to efficiently explore the co-observation schedules of many MAPF continuations from the current $\alpha(t)$ simultaneously. Although our abstraction does over-approximate the set of MAPF continuations, we can prove that the abstractions preserve the possibility of pairwise co-observation. That is, if under the abstraction it is possible for a robot i to observe robot j at a location v at time t, then there is some MAPF continuation where j is observed at location v at time t. This property of the abstraction ensures that Alg. 1 is sound, since Alg. 1 is essentially verifying that there is no forbidden deviation through the complement of the observed region under the non-deterministic movement abstraction.

The input to Alg. 1 is the centralized MRS instance $G=(V,E)$, $S=(V,E^*)$, $V_{\text{forbidden}}(t)$, and the sequence of announcements planned by the CE from the current time t to a future time v, $\{\alpha(s)\}_{s\in[t,v]}$. We iteratively fix each robot $a \in R$ as the compromised robot; by attacker independence, from a's perspective the other $R\setminus a$ may all be non-compromised. We now iterate over the $s \in [t,v]$ and attempt to verify that $\alpha(s)$ is not informative enough to reveal a forbidden and undetected plan-deviation attack for robot a beginning at time s. Verifying $\alpha(s)$ has two phases: (1) compute the soonest time $u^* > s$ and a location l_{obs} where a could be observed by a robot in $R\setminus\{a\}$ and (2) show that no forbidden undetected deviation exists for a on (s,u^*).

Since $\alpha(s)$ only reveals partial planning information for $i \in R$ up to time $|\alpha(s)^i|$, we account for the unknown future of a given robot by allowing them to move non-deterministically on G for time steps $u > |\alpha(s)^i|$. We denote the set of locations that $i \in R$ (non-)deterministically occupies at time u as X^i_u. The dynamics of the non-deterministically-moving agents are as follows:

1. For all $i \in R$, $X^i_u = \{\alpha(s)^i_u\}$ for $u \leq |\alpha(s)^i|$, i.e. robots move deterministically for

times where their position is specified by $\alpha(s)$.

2. For $u > |\alpha(s)^i|$, $X_u^i \leftarrow \mathcal{N}_G(X_{u-1}^i)$, i.e. non-deterministically-moving robots follow all edges in G from the set of locations previously occupied.

3. For $u > |\alpha(s)^i|$, remove from X_u^i all locations that are deterministically occupied by other robots $R \setminus \{i\}$, or would lead to a vertex- or edge-conflict with a deterministically-moving robot in $R \setminus \{i\}$. The conflict locations are stored in a set C, which is updated with a new conflict whenever there is a $c \in X_{u-1}^i$ that has no children (available actions) to X_u^i. The verification for $\alpha(s)$ is restarted whenever a new conflict is found.

4. The non-deterministically-moving compromised robot a cannot move into any location *previously* occupied by non- deterministically moving robots in $R \setminus \{a\}$, so remove from X_u^a all elements also in $X_{u-1}^{R\setminus\{a\}}$.

5. Non-deterministically-moving robots in $R \setminus \{a\}$ cannot move into any location occupied non-deterministically by a, so remove from $X_u^{R\setminus\{a\}}$ all elements also in X_u^a.

6. The non-deterministically-moving compromised robot a cannot move into any location occupied by non-deterministically-moving robots in $R \setminus \{a\}$, so remove from X_u^a all elements also in $X_u^{R\setminus\{a\}}$.

The non-deterministic dynamics are evolved for $u = s+1, \ldots, u^*$, where u^* is the first time step s.t. $\exists l_{\text{obs}} \in X_{u^*}^a$ s.t. $l_{\text{obs}} \in \mathcal{N}_S(X_{u^*}^{R\setminus\{a\}})$, i.e. when a possible observation on a by another robot in $R \setminus \{a\}$ is found, concluding the first phase of verifying $\alpha(s)$. For the second phase, we simply check via graph search on G from source vertex $\alpha(s)_s^a$ if there is a MAPF deviation $\tilde{x}$ for a on (s, u^*) s.t. for all $u \in (s, u^*)$, $\tilde{x}_u^a \notin \mathcal{N}_S(X_u^{R\setminus\{a\}})$. If no such deviation is found, then we return `true`, signifying that there exists a continuation from $\alpha(s)$ s.t. no forbidden undetected MAPF deviation exists for a on (s, u^*) (in that continuation). If each $\alpha(s)$ is verified for each $i \in R$, then the announcements $\{\alpha(s)\}_{s\in[t,v]}$ are verified to be horizon-limiting MAPF announcements.

Theorem 3 (Soundness of Non-Deterministic Co-Observation Enumeration). *Let x be a MAPF plan and α an announcement sequence for x. Then if Alg. 1 returns* `true`*, then α is a horizon-limiting MAPF announcement for x.*

Proof. Eq. 4.1 is equivalent to the statement that for all forbidden MAPF deviations $\tilde{x}$ for $a \in R$ on $(s, s+k)$, there exists a MAPF continuation y of $\alpha(s)$ s.t. execution of attack $\tilde{x}$ would trigger a co-observation-based detection if y is the CE's MAPF plan. Let $\{X_u\}_{u>s}$ be the sequence of (non-) deterministically reachable sets for the robots starting at time s as computed by Alg. 1. We begin by proving a lemma that the non-deterministic movement abstraction of Alg. 1 is sound w.r.t. possibility of co-observation:

Lemma 1 (Non-deterministic Abstraction Preserves Possibility of Co-observations). *Let $q \in X_u^a$. If $q \in \mathcal{N}_S(X_u^{R\setminus\{a\}})$, then there exists a MAPF continuation y, $y \succeq \alpha(s)$ s.t. $y_u^a \in \mathcal{N}_S(y_u^{R\setminus\{a\}})$. In other words, if it is possible under the non-deterministic abstraction for a to be observed at time u at location q, then there exists a MAPF continuation from $\alpha(s)$ where a is observed at time u at location q.*

Proof of Lem. 1: Firstly, since for all $i \in R$ there are no elements of X_u^i that are not in $\mathcal{N}_G(X_{u-1}^i)$, we have that all locations in X_u^i are reachable in one time step by taking an edge in E from some location in X_{u-1}^i. Furthermore, since elements in X_u^i are not in the conflict set C, we have that there is a conflict-free walk on G from $\alpha(s)_s^i$ to each element in X_u^i w.r.t. the known prefix $\alpha(s)$. Since non-deterministically-moving a is not allowed to move into any element in $X^{R\setminus\{a\}}$, we therefore have that for all $q \in X_u^a$, there exists a $y \succeq \alpha(s)$ s.t. $y_u^a = q$. As for non-deterministically-moving pairs of other robots in $R \setminus \{a\}$, the abstraction does not explicitly prevent vertex- and edge- conflicts. However, since elements in $X_u^{R\setminus\{a\}}$ are not in the conflict set C, we have that it is possible for some robot $i \in R \setminus \{a\}$ to occupy each element of $X_u^{R\setminus\{a\}}$ without causing a conflict with deterministically-moving robots. Similarly, since non-deterministically-moving robots in $R \setminus \{a\}$ are not allowed to move into any element in X^a, we conclude that the only conflicts preventing a robot $i \in R \setminus \{a\}$ from reaching a location in X_u^i is a conflict with a different non-deterministically-moving robot $j \in R \setminus \{a, i\}$. Therefore, for all $p \in X_u^i$, either there exists a $y \succeq \alpha(s)$ s.t. $y_u^i = p$ or there exists a $y \succeq \alpha(s), j \in R \setminus \{a, i\}$ s.t. $y_u^j = p$. We conclude that for all $q \in X_u^a, p \in X_u^{R\setminus\{a\}}$, there exists a $y \succeq \alpha(s)$, $i \in R \setminus \{a\}$ s.t. $y_u^a = q$ and $y_u^i = p$. ■

Proof of Thm. 3: Now let u^*, l_{obs} be the time and position of the first possible observation on a as computed by Alg. 1. Let $\tilde{x}$ be any forbidden MAPF deviation for a on $(s, s+k)$,

$k > 1$. *Case I, $\tilde{x}^a_{u^*} \neq l_{obs}$*: it follows immediately from Lem. 1 that $\exists y \succeq \alpha(s), i \in R \setminus a$ s.t. $l_{\text{obs}} \in \mathcal{N}_S(y^i_{u^*})$. Therefore, a misses an observation, triggering a co-observation-based detection in the CE. *Case II(a), $\tilde{x}^a_{u^*} = l_{obs}$ and $\exists u \in (t, u^*)$ s.t. $\tilde{x}^a_u \in \mathcal{N}_S(X_u^{R\setminus\{a\}})$*: in this situation, it again follows immediately from Lem. 1 that $\exists y \succeq \alpha(s), i \in R \setminus \{a\}$ s.t. $\tilde{x}^a_u \in \mathcal{N}_S(y^i_u)$. However, since $u < u^*$ we contradict that the first possible observation on a occurs at time u^*. Therefore, a has caused an unexpected observation, triggering a co-observation-based detection in the CE. *Case II(b), $\tilde{x}^a_{u^*} = l_{obs}$ and $(\neg\exists u \in (t, u^*)$ s.t. $\tilde{x}^a_u \in \mathcal{N}_S(X_u^{R\setminus\{a\}})$*: the only remaining forbidden deviations are those where a does not miss the planned observation at time u^*, and does not introduce an unexpected observation at times $u \in (t, u^*)$. The algorithm performs a graph search to check that no such deviation exists. ■ □

The computational complexity of Alg. 1 is $O(R^2V)$.

4.5 Experimental Results

In the preceding section, we have proposed a strategy for mitigating plan-deviation attacks that rests on robot co-observations and on limiting how much planning information is revealed at any given moment. Here, we seek to answer the following research questions:

RQ1 What is the security benefit of HoLA, compared to a centralized MRS that detects problems using localization self-reports only?

RQ2 Compared to a non-security-aware centralized MRS, what is the overhead of HoLA?

RQ3 What properties of robot co-observations from general MAPF plans lead to security vulnerabilities?

RQ4 What is the security benefit for robot co-observation? How does the inclusion of robot co-observations impact our ability to mitigate plan-deviation attacks without using horizon-limiting announcements?

RQ5 How does the announcement schedule for incremental plans impact the effectiveness of attacks?

RQ6 What is the security vulnerability associated with announcement schemes that may be used in typical centralized MRS deployments?

4.5.1 Experimental Setup

Environment. MAPF plans are computed using the ECBS algorithm [Barer et al., 2014b], an efficient and bounded sub-optimal graph-based MAPF solver (and so, applicable for centralized MRS), for a set of 100 standard MAPF 4-connected grid benchmark instances [Hönig, 2021]. The MAPF instances are solvable (i.e. there exists a MAPF plan that solves the instance), randomly generated 4-connected 32×32 grids with either 10, 20, . . . , or 100 robots and $\sim$200 obstacles. We assume each robot has sensing capability within adjacent squares, that is the sensor model for each robot, S, is the same as the reachability graph G. Robots are assumed to mutually co-observe each other if they are adjacent on the grid. We implement the announcement security verification in the Rust programming language; runtimes are reported on an Intel Core i7-6700 processor at 4GHz. Source code to reproduce our experiments can be found at `https://github.com/gitsper/hola-announce`

Execution scenarios. There are two factors that influence the announcement schedules: 1. how many steps ahead are included in the announcement and 2. how many announcements are sent in a communication from the CE to the robots. The number of steps ahead represent a trade-off between security and delay in computing the task, for increased security the announcement should include only one step but this will results in increased time in completing the task by the team of robots. We use the following notation:

- (p,k)-announcements: specify that the CE makes a new announcement every p timesteps and each announcement includes planning information for the next k steps.

Stealthy attacker metrics. We consider the security of a nominal execution scenario compromised by a stealthy attacker. Instead of performing simulations, for each scenario we randomly pick one of 10 different robots to play the role of stealthy attackers and one of

10 locations in the grid to be marked as the forbidden location and use Alg. 1 to check if the announcements are horizon-limiting w.r.t. the compromised robots and forbidden location. We use the following metric:

- *Secure stealthy scenario* is the proportion of scenarios that can be verified by Alg. 1 as secure from the stealthy attacker given a set of possible scenarios and is an indicator of how vulnerable the case is to stealthy attackers.

A MAPF instance with associated ECBS-computed MAPF plan, co-observation schedule, and announcement schedule make up an *execution scenario*, or scenario for short.

Bold attacker metrics. In each scenario, we additionally examine the behavior of a non-stealthy, or bold, attacker that may attempt an attack even if it is not sure the attack will be a forbidden and undetected deviation. The behavior of the bold attacker is deviate to $V_{\text{forbidden}}$, matching the co-observation schedule as well as possible given the information in $\alpha(t)$ without reasoning about eventual continuations from $\alpha(t)$. We simulate a bold attacker 100 times, each time randomly picking one of 10 different robots to play the attacking role and one of 10 locations in the grid to be marked as the forbidden location. We average the metrics over the scenarios. For bold attackers, we cannot verify that the plans are secure, so the CE will attempt to detect, but we cannot guarantee detection. We use the following metrics to capture the attacks and their detection:

- *Bold attack success* is the proportion of simulations where the compromised robot performs a forbidden deviation and is an indicator of how relatively dangerous the compromised robot is in the set of scenarios.

- *Bold detection miss* is the fraction of positive cases where the CE reports no anomaly based on our self-report-based detection mechanism and is an indicator of how many forbidden deviations are missed by the CE.

4.5.2 Security Benefit of HoLA

The central thesis of this paper is that robot self-reports of localization alone simply do not suffice to detect or prevent malicious behavior in centralized MRS. As such, the primary contribution of our paper is HoLA, a security measure for the CE leveraging robot co-observations and horizon-limiting announcements. With HoLA, the CE can compute and release maximal announcements such that there is a guarantee that all stealthy attacks in the system will be prevented. Furthermore, HoLA aids in the detection of non-stealthy attackers in the system by simultaneously making the attack-planning problem more difficult and by gathering the co-observation reports. To demonstrate the necessity of HoLA (**RQ1**), we consider a bold attacker and we compare two CE implementations, 1. with localization-based detection only that releases the full MAPF plan to the robots (no mitigation) and 2. HoLA: co-observation-based detection where the CE releases maximal-length announcements that have been verified with Alg. 1 as preventing all stealthy attacks.

In Fig. 4·2, we show the miss detection for the bold attacker as a function of $|R|$, the number of robots in the execution scenario. We observe that when the CE employs no mitigation, essentially all forbidden deviations by the bold attacker are missed by the CE, highlighting the inadequacy of localization-based detection. Whereas with HoLA, not only

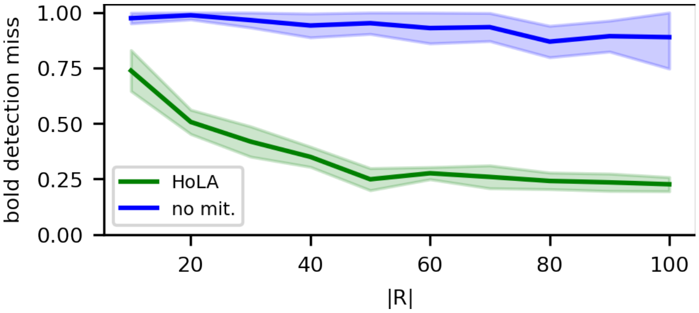

Figure 4·2: Bold detection miss for the bold attacker when the CE employs either localization-based detection only (no mitigation) or HoLA. In the HoLA case, the CE collects co-observation reports and the announcements are of maximal length that are verified as horizon-limiting.

are all stealthy attacks provably prevented, the CE misses far fewer bold attacks, ultimately reaching a bold detection miss of just 22% for $|R| = 100$. HoLA consistently outperforms the CE with no mitigation in terms of detection; bold detection miss is lower for larger $|R|$ due to more frequent co-observations in more congested environments. We note that for certain scenarios, the bold attacker is certain to succeed without being detected by HoLA, e.g. when the compromised robots are close to the forbidden zone and far away from other robots.

4.5.3 Overhead of Announcement Security Verification

Our solution proposes that the CE should use Alg. 1 to verify that the chosen α are horizon-limiting MAPF announcements. The verification procedure has a computational overhead that depends on the number of robots, $|R|$. Our scenario set had instances between 10 and 100 robots with a maximum MAPF length of 70 time steps (the average length is 49 time steps). Across all scenarios, we verify each robot in sequence using Alg. 1; it never took longer than 6 minutes to terminate. For $|R| = 10$, the average time was 48 sec. and for $|R| = 100$, 2.11 min. As the announcements are verified for each robot independently, the computation is *parallelizable*. As a point of comparison, MAPF instances in our benchmark took ECBS up to 1 min. to plan (with suboptimality bound 1.3), whereas on our 8 core CPU the verification procedure took up to 6 min./8 = 45 sec.

4.5.4 Security of General MAPF Plans

We aim to understand what qualities of general MAPF plans contribute to or detract from the security of the scenario under HoLA (**RQ3**). From the perspective of the attacker, what makes an deviation-detecting plan secure is that there does not exist an undetected forbidden plan-deviation between consecutive observations made on the attacker. We perform the following experiment: we ran attack scenarios with a simulated bold attacker where we varied the number of ahead steps included in announcement and we organize the scenarios

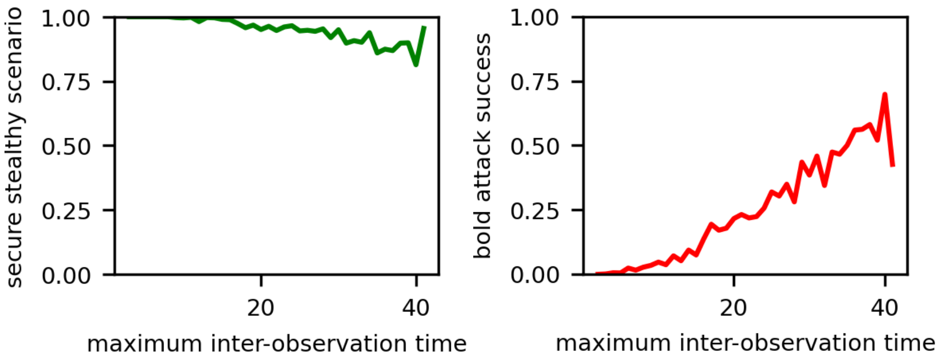

Figure 4·3: Attacker success for a bold attacker and secure scenarios for a stealthy attacker for *general MAPF plans* (i.e. plans not known if they are deviation-detecting because they were not generated as such), as a function of the time that robots go unobserved (maximum inter-observation time).

by the maximum amount of timesteps the attacker has between consecutive observations, we refer to this as *maximum inter-observation time*.

In Fig. 4·3, we plot the bold attack success and observe that attackers that have fewer (< 15) timesteps at most between consecutive observations have attack success ratio below the 10% whereas attackers that have large gaps between consecutive observations (> 25) have attack success significantly above average, reaching a attack success of $\sim 70\%$ at maximum inter-observation times of 40 timesteps. The correlation between increased maximum inter-observation time and worsened security is also confirmed by tracking the secure scenarios for the stealthy attacker verification attempts: we find that attackers that are observed at least every 10 timesteps have 100% secure scenarios, beyond which point the secure scenarios trend downward ultimately reaching 81% for the least-observed stealthy attackers.

4.5.5 Ablation Study: Security Benefit of Robot Co-observations

In Section 4.4.1 we claimed that robot self-reports containing only localization information are not sufficient to provide security guarantees. We support through experimental results this claim (**RQ4**). We consider a bold attacker and we compare two CE imple-

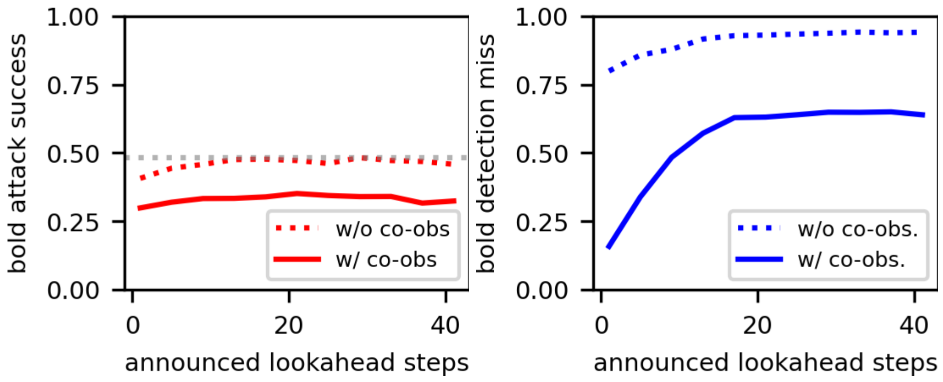

Figure 4·4: Attack success and miss detection for the bold attacker, with co-observations enabled and without co-observations, where the CE uses a $(1,k)$-announcement schedule; where k represents the number of lookahead steps. In some scenarios the attack is not feasible, resulting in upper bound bold attack success marked with gray dotted line.

mentations, one with localization-based detection only and one with co-observation-based detection, where we vary the amount of information available in each announcement, i.e. how many steps ahead are included in the announcement. In order to trigger a detection by the CE in the case where no co-observations are used, the compromised robot would need to either collide with a non-compromised robot or otherwise occupy the location that a non-compromised robot is meant to occupy. For the case with co-observations, a detection would also occur if the reported co-observations do not match what the CE expected (see Def. 17).

In Fig. 4·4 we show the attack success and the miss detection for the bold attacker, as a function of k, the number of lookahead steps included in an announcement. We observe that in the situation with minimal announcements ($k = 1$), for the no co-observation setting, the CE has a miss ratio of 80% whereas with the robot co-observations present the CE has a miss ratio of just 16%. As the announcements become more informative (k larger than 25), the miss ratio of the no co-observation CE approaches almost 94% whereas the CE that gathers co-observations approaches a miss ratio of 64%. The bold attacker success decreased from about 46% to 33% in the cases when no co-observations are used, or when co-observations are used, respectively.

4.5.6 Ablation Study: Impact of the Announcement Schedule on Security

We have argued in Section 4.4.2 that announcement schedules impact the security of a scenario, since more informative announcement schedules decrease the set of plan deviations that the attacker considers to be possibly forbidden and undetected (**RQ5**). However, we have no theoretical guarantee that (1) Alg. 1 is able to prove more of the less informative scenarios to be horizon-limiting and (2) that the theoretical increase in attack planning difficulty for bold attackers under less informative announcements corresponds to a measurable decrease in attacker success and stealth.

We organize the results in Fig. 4·5 by the parameter k, the number of lookahead steps in an announcement; by monotonicity of announcements, $(1,k)$-ann. are less informative than $(1,k+1)$-ann., and (k,k)-ann. are less informative than $(1,k)$-ann. For the stealthy attacker security verification (see Fig. 4·5), we indeed observe a negative correlation between k and the secure scenarios: e.g. across all $(1,k)$-ann. scenarios we have secure scenarios of approx. 98% for minimal ($k=1$) announcements, which drops to secure scenarios of approx. 90% as k increases. We further report that the density of the agents in the environment has a large impact on how quickly our ability to verify security with Alg. 1 deteriorates with k. As an example, across the scenarios with few robots, $|R|=10$, the secure scenarios drops to approx. 70% whereas for scenarios with $|R|>70$ the secure scenarios does not drop below 95%. In addition to supporting that our proposed approach is able to verify the security of a majority of scenarios w.r.t. stealthy attackers, we note that increased density of non-compromised robots improves our ability to verify the security of the system.

We observe a positive correlation between the lookahead parameter k of (p,k)-announcements and the miss ratio of the CE detections for the bold attacker (Fig. 4·5). Specifically, minimal

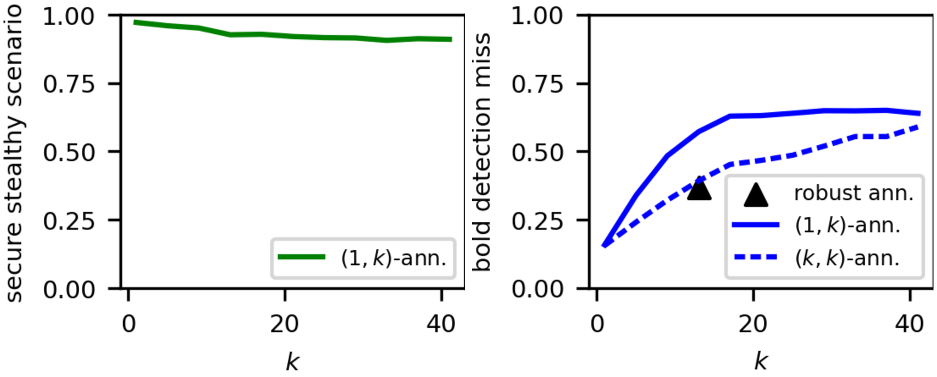

Figure 4·5: Secure scenarios for the stealthy attacker and missed detection for a bold attacker, comparing scenarios where the CE uses $(1,k)$-announcements or (k,k)-announcements; announcing the next k steps at every time step and announcing the next k steps once every k steps respectively. We also plot the detection miss against the average lookahead for the robust announcement scheme.

announcements ($k = 1$) correspond to miss ratios of less than 20%, but with the most informative announcements tested the miss ratio approaches up to approx. 60%. The results indicate that even though limiting the announcements does not change the set of behaviors available to bold attackers, in practice the increase in ambiguity the attacker experiences when choosing a plan deviation has a significant impact on the stealth of the bold attacker.

The synthetic announcement classes help to demonstrate the relationship between information release and security, however the synthetic announcement classes are not directly comparable to other announcement schemes that may be used in a typical MRS deployment (**RQ6**). As a point of comparison, we consider scenarios where robust announcements are computed from the motion plan as per [Honig et al., 2019]. Since the robust announcements have dynamic prefix lengths that are different for each robot, we measure the average per-robot prefix length, that is across all plans, times, and robots, the typical amount of future planning information that is available about a given robot. We find the average lookahead to be approx. 13 timesteps, and that for the robust announcement scheme the CE has a miss ratio of approx. 37% (Fig. 4·5). The results indicate that from a security perspective, the robust announcement scheme is quite similar to the (k,k)-announcement class (since $(13,13)$-announcements have a similar miss ratio).

4.6 Summary

In this chapter we focused on the problem of mitigating *plan-deviation attacks* with robot co-observations and incremental plan release. The attacker has two goals: first, to move toward a forbidden zone, and second, to remain undetected by the central entity. We leverage co-observation to mitigate the ability of the attacker to lie about its location; and we limit the size of the incremental plan announcements so that the attacker has limited ability to confidently plan ahead. We describe two types of attackers – "stealthy", and "bold" – based on their desire to remain undetected or not. We prove that our solution prevents attacks for a set of stealthy attackers. For bold attackers we show experimentally that our solution significantly increases the detection of the attacks. Our solution also has a small overhead making it practical for sets of tens to hundreds of robots.

Algorithm 1 Non-deterministic Co-observation Enumeration

```
procedure VERIFY($G, S, V_{\text{forb.}}, \alpha, s, a$)
    $u \leftarrow s$                                                    ▷ time offset
    $X_u^R \leftarrow \alpha(s)_u^R$                                    ▷ init. reachable sets for each robot
    do
        $X_{u+1}^R \leftarrow$ REACHABLE($\alpha(s), G, X_u^R, u, C$)
        $X_{u+1}^a \leftarrow X_{u+1}^a \setminus X_u^{R\setminus\{a\}}$
        $X_{u+1}^{R\setminus\{a\}} \leftarrow X_{u+1}^{R\setminus\{a\}} \setminus X_{u+1}^a$
        $X_{u+1}^a \leftarrow X_{u+1}^a \setminus X_{u+1}^{R\setminus\{a\}}$
        $u \leftarrow u + 1$
    while $X_u^a \cap \mathcal{N}_S(X_u^{R\setminus\{a\}}) \neq \{\}$
    $u^* \leftarrow u$
    $Q \leftarrow X_{u^*}^a \cap \mathcal{N}_G(X_{u^*}^{R\setminus\{a\}})$
    return $\bigvee_{q\in Q} \neg$ATTACKEXISTS($a, G, V_{\text{forb.}}, X, s, u^*, q$)
end procedure
procedure REACHABLE($x, G, X, t, C$)
    $X_{\text{next}} \leftarrow \{\}$
    for $v \in X$ do
        $X_{\text{next}} \leftarrow X_{\text{next}} \cup$ MOVEROBOT($x, G, v, t, C$)
    end for
    return $X_{\text{next}}$
end procedure
procedure MOVEROBOT($x, G, v, t, C$)
    if $\exists r \in R, v = x_t^r \wedge |x^r| > t + 1$ then
        return $\{x_{t+1}^r\}$                                          ▷ prefix for $r$ is known
    end if
    ret $\leftarrow \mathcal{N}_G(v) \setminus \{x_{t+1}^r : r \in R \wedge |x^r| > t + 1\}$
    if $v \notin$ ret then                                              ▷ avoid edge conflict
        ret $\leftarrow$ ret $\setminus \{x_t^r : r \in R \wedge x_{t+1}^r = v\}$
    end if
    ret $\leftarrow$ ret $\setminus \{v : (v, t+1) \in C\}$
    if |ret| $= 0$ then
        $C \leftarrow C \cup \{(v, t)\}$
        raise CONFLICT
    end if
    return ret
end procedure
procedure ATTACKEXISTS($a, G, V_{\text{forb.}}, X, s, u^*, q$)
    A $\leftarrow X_t^a$
    B $\leftarrow \{\}$
    for $u = s + 1, \ldots, u^*$ do
        A $\leftarrow \mathcal{N}_G(A) \setminus \mathcal{N}_S(X_u^{R\setminus\{a\}})$
        B $\leftarrow \mathcal{N}_G(B) \setminus \mathcal{N}_S(X_u^{R\setminus\{a\}})$
        B $\leftarrow$ B $\cup$ (A $\cap V_{\text{forb.}}$)
    end for
    return $q \in B$
end procedure
```

Chapter 5

Decentralized Blocklist

As mentioned in Section 2.1.2, emergent MRS applications in unstructured environments are not amenable to centralized approaches due to communication constraints [Gielis et al., 2022]. Given the multitude of possible attacks, it is important to understand the resilience of the MRS from Byzantine attackers – that is if an unknown, non-empty, and proper subset of the robots is allowed to have arbitrarily different behaviors relative to the cooperative robots in terms of physical actions and communication.

Byzantine-unaware MRS implementations are often highly vulnerable, and break completely, when even one robot has been comprised. In our case studies for example, Byzantine robots may cause robots within a swarm to follow a false target, or have arbitrarily large errors in time synchronization or localization. Two main approaches have been proposed for Byzantine-resilient MRS: 1. distributed ledger technology [Strobel et al., 2018,Pacheco et al., 2020,Aditya et al., 2021] and 2. the Weighted-Mean Subsequence Reduced (W-MSR) algorithm [LeBlanc et al., 2013,Saulnier et al., 2017,Mitra et al., 2019]. Distributed ledger approaches have been investigated for a diverse range of applications, but the associated compute time and energy usage overhead, and the necessity for application-specific fine-tuning limit their practicality. On the other hand, W-MSR is easy to implement and has well-understood theoretical guarantees. However, W-MSR can only be used for MRS applications that are implemented via Linear Consensus Protocol (LCP), performance does not scale with the number of robots in the system, and the number of Byzantine robots to tolerate, F, is a parameter that must be known a priori. Suppose that LCP is the means by which the robots reach a collective decision about a physical property of the environment. The choice of F in W-MSR dictates how many outliers robots should discard in each update of linear consensus; each robot needs at minimum $2F+1$ cooperative neighbors in order to update their local consensus variable and at minimum $F+1$ cooperative robots must independently make direct measurement of the underlying physical quantity. If F is chosen smaller than the number of Byzantine robots, then the mitigation provided by W-MSR is forfeit. For large F, the network connectivity requirement and the logistics of maintaining $F+1$ cooperative observers renders W-MSR impractical.

In this work we propose acdbp, an approach to Byzantine resilience inspired by P2P networks, based on inter-robot accusations. Cooperative robots make use of co-observations to detect misbehaving peers and make accusations accordingly. Accusations propagate through the cooperative nodes and are fed to a local matching algorithm that outputs a blocklist. We derive necessary and sufficient conditions on the set of accusations that must be made and connectivity of the MRS that ensures that all Byzantine robots are eventually blocked by the cooperative robots, and their influence mitigated. Specifically, we show that for a closed MRS satisfying an analogous $(F+1)$-connectivity requirement for time-varying networks, blocking all of the Byzantine robots is equivalent to Hall's marriage condition on the accusations made within the system. In addition to W-MSR requiring the number of Byzantine robots to tolerate be known a priori, we claim that W-MSR does not scale with the number of robots in practice. We show empirically on target tracking and time synchronization applications that this is the case, and that our proposed approach adaptively scales to hundreds of robots/attackers, in contrast to just one or two attackers in a swarm of no more than 20 robots as in related works. W-MSR cannot be used to provide Byzantine resilience for MRS not implemented over LCP, such as cooperative localization. We will implement Byzantine-resilient cooperative localization using our approach as a proof of concept; to our knowledge ours is the first successful technique for decentralized

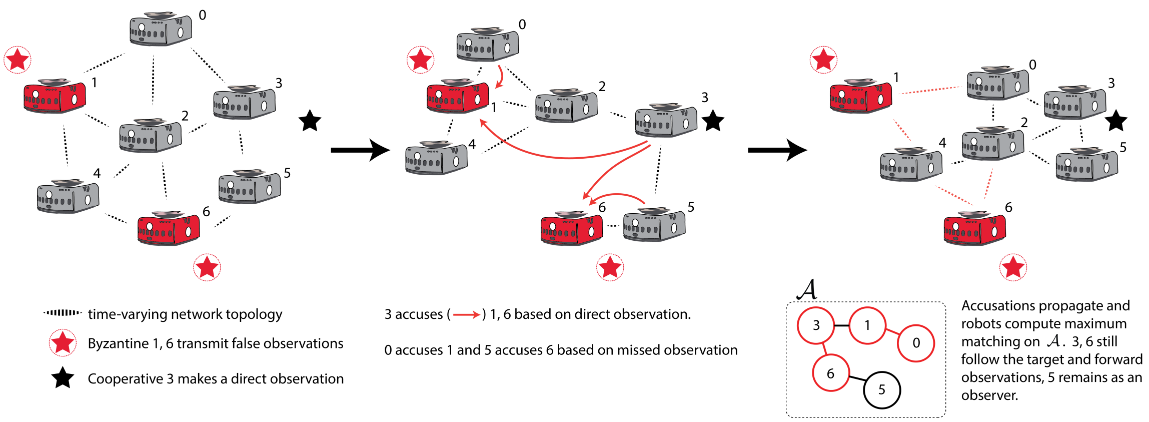
0
1
2
3
4
5
6
time-varying network topology
Byzantine 1, 6 transmit false observations
Cooperative 3 makes a direct observation
3 accuses (⟶) 1, 6 based on direct observation.
0 accuses 1 and 5 accuses 6 based on missed observation
$\mathcal{A}$
Accusations propagate and robots compute maximum matching on $\mathcal{A}$. 3, 6 still follow the target and forward observations, 5 remains as an observer.

Figure 5·1: DBP is used to provide Byzantine resilience for a simple seven robot (two Byzantine) scenario of the target tracking case study explored in Section 5.4.1. Byzantine robots 1 and 6 transmit false target observations (red stars) while cooperative robot 3 makes a direct observation of the target (black star). Based on the closest observer, the cooperative robots move towards the supposed target location. Robots 0 and 5 make it to the false locations reported by 1 and 6 respectively and accuse the observers on the basis of missed observations that should have been made. Meanwhile, robot 3 accuses both 1 and 6 since their observations contradict 3's direct observation. The accusations flood through the cooperative robots, until each cooperative robot has accusation graph denoted $\mathcal{A}$. Edmond's algorithm is used by each robot to compute the maximum matching $\{(3,6),(0,1)\}$, thus observations from robots 0, 1, 3, and 6 are blocked. Observations from robots 2, 4, and 5 are still trusted by the other cooperative robots. Note that 0 and 3 still continue to cooperate by moving towards the target and forwarding observations from non-blocked observers.

and Byzantine-resilient cooperative localization.

5.1 Additional Background & Related Work

W-MSR. Perhaps the most well-understood approach to Byzantine-resilient decentralized MRS is the W-MSR algorithm. W-MSR can be applied to MRS applications that are implemented over Linear Consensus Protocol – a distributed consensus algorithm for real-valued variables whereby in each timestep robots update their local variable to a convex combination of their neighbor's broadcast values, i.e.

$$x_i(t) = \sum_{j \in \mathcal{N}_G(i)} \alpha_j x_j(t-1) \text{ where } \sum \alpha_j = 1$$

and $\mathcal{N}_G(i)$ are the neighbors of i in the connectivity graph G. The authors of [LeBlanc et al., 2013] first introduced W-MSR for Byzantine resilience which discards the F highest and F lowest values received at each timestep of LCP, and show that convergence despite up to F Byzantine robots is equivalent to a graph robustness property. Specifically, if the connectivity graph of the robots is at least $(2F+1)$-vertex-connected, then the consensus will converge to a value within the convex hull of the cooperative robots' initial values. W-MSR has been applied to a variety of applications, such as flocking [Saulnier et al., 2017] and state estimation [Mitra et al., 2019]. Extensions for the W-MSR algorithm to time-varying networks where the union of the connectivity graphs within a bounded window is robust are proposed in [Saldaña et al., 2017] and to event-driven control in [Amirian and Shamaghdari, 2021]. Methods to form robust graph topologies, as required by the W-MSR algorithm, are proposed in [Guerrero-Bonilla et al., 2017].

5.2 Threat Model

We consider a swarm robotics system with robots connected by time-varying network topology $G[t] = (V, E[t])$. Each robot has an identity that has been issued by a trusted central authority at deploy-time, which it uses to both send signed messages to its neighbors and to verify the authenticity of received messages. Assume that some unknown subset of the robots have been compromised by a Byzantine adversary. We refer to the cooperative robots as $\mathcal{C} \in V$, the Byzantine ones as $\bar{\mathcal{C}} = V \setminus \mathcal{C}$, and we assume that the sets V and $\mathcal{C}$ are fixed, i.e. the MRS is *closed*. We assume a strong adversary, where the Byzantine robots can coordinate centrally with each other online, have detailed knowledge of the system implementation such as robot capabilities and application details, can send arbitrary messages to the cooperative robots, and have arbitrary physical behaviors. The goal of the Byzantine robots is to disrupt the MRS application; the specific goal and attack strategy will depend on the application. Each of our case studies will specify the attacker's goal and strategy. Since the robots use their trust authority-issued identities to communicate, we assume that Sybil attacks are not possible for the adversary, since the adversary is unable to forge fraudulent identities that will be accepted by $\mathcal{C}$. However, robots whose identities (secret keys) have been compromised can be used by the adversary to send misleading messages. Therefore, any robots in the swarm whose keys have been compromised are considered to be part of $\bar{\mathcal{C}}$.

5.3 Decentralized Blocklist Protocol

DBP is a swarm blocklist algorithm that is adaptive to the presence of Byzantine adversaries. DBP can be used as an alternative to W-MSR, but with lower requirement on network connectivity and without needing to know F ahead of time. DBP is adaptive, and as such the requirement on robust network topology scales with the true number of Byzantine robots. The connectivity requirements of W-MSR scale with the parameter F, even if the actual number of Byzantine robots is lower. An example of how DBP works on a target tracking scenario is shown in Fig. 5·1. Based on locally-made observations, cooperative robots accuse misbehaving peers. The accusations propagate through the network via flooding and are used as input to a matching algorithm that outputs a blocklist.

DBP relies on flooding as a networking primitive, where cooperative robots always rebroadcast (forward) received messages. Messages in DBP are accusations signed by the robot initiating the flood. Accusations $\texttt{Acc}_i(j)$ are an application-agnostic message and the payload is simply the identity of a robot j that the origin i wishes to accuse. The precise rules used to decide if and when an accusation should be issued are application-specific. Accusations serve to remove the influence of Byzantine nodes on the swarm application.

Each robot i locally maintains a set $R_i[t]$ of accusations that it has received. A subset $R_i^*[t] \subseteq R_i[t]$ will be locally computed by i using any deterministic maximum matching algorithm (such as Edmond's [Edmonds, 1965]) to form the blocklist. For the remainder of this section, we will assume that the robots have a *sound* accusation mechanism:

Definition 20 (Sound accusation)**.** *If a cooperative robot i issues an accusation* $\texttt{Acc}_i(j)$*, then* $\texttt{Acc}_i(j)$ *is* sound *if and only if* $j \in \bar{\mathcal{C}}$*.*

Remark 1. *In the presence of Byzantine robots, receiving a message* $\texttt{Acc}_i(j)$ *implies that* $i \in \bar{\mathcal{C}} \vee j \in \bar{\mathcal{C}}$*. The reason is that if* $i \in \mathcal{C}$*, then* $j \in \bar{\mathcal{C}}$ *by soundness of accusations. In the other case,* $i \in \bar{\mathcal{C}}$*.*

Matching. Importantly, the set of received accusations $R_i[t]$ has a structure imparted by the accusation soundness. Given an undirected graph $G = (V = X \cup Y, E)$ with X, Y disjoint, we say that G is *X-semi-bipartite* if X is an independent vertex set in G. A subset $\mathcal{M} \subseteq E$ is a *matching* on G if $\mathcal{M}$ is an independent edge set in G. Given a matching $\mathcal{M}$, we denote by $V_{\mathcal{M}}$ the matched vertices in $\mathcal{M}$. If no additional edges can be added to a matching $\mathcal{M}$, then $\mathcal{M}$ is *maximal*. If there does not exist a matching $\mathcal{M}^*$ s.t. $|V_{\mathcal{M}^*}| > |V_{\mathcal{M}}|$, then $\mathcal{M}$ is a maximum cardinality, or *maximum*, matching. Given a subset $S \subseteq V$, a matching $\mathcal{M}$ is *S-perfect* if $S \subseteq V_{\mathcal{M}}$. The following condition allows us to connect the notion of maximum matching and perfect matching:

Definition 21 (Hall's Marriage condition)**.** *Given* $G = (X \cup Y, E)$ *s.t.* G *is* X*-semi-bipartite, a* Y*-perfect matching exists if* $\forall S \subseteq Y$*,* $|S| \leq |\mathcal{N}_G(S) \cap X|$*. Additionally, any maximum matching will be* Y*-perfect.*

Accusation graph. Now let $\mathcal{A}_k[t]$ be the *accusation graph* with edge (i, j) iff $\texttt{Acc}_i(j) \in R_k[t]$. As we note in Remark 1, each accusation can be viewed as a disjunction – $\texttt{Acc}_i(j)$ can be understood as "*i* is Byzantine or *j* is Byzantine (or both are)." Therefore, $\mathcal{A}_k[t]$ is $\mathcal{C}$-semi-bipartite, and any matching M on $\mathcal{A}_k[t]$ will satisfy $|V_M| \leq 2|\bar{\mathcal{C}}|$. The inequality will be tight if and only if the Hall marriage condition holds for $\bar{\mathcal{C}}$ on $\mathcal{A}_k[t]$ – in which case the maximum matching M is $\bar{\mathcal{C}}$-perfect with $|V_M| = 2|\bar{\mathcal{C}}|$. Robot k chooses $R^*_k[t]$ to be the matched vertices of the maximum matching on $\mathcal{A}_k[t]$ – the robots corresponding to the matched vertices are the ones that k will block. An example accusation graph and associated maximum matching is shown as "$\mathcal{A}$" in Fig. 5·1.

Network flooding. This matching result is only useful if the requisite accusations actually propagate through the robots in $\mathcal{C}$. Given a time-varying directed graph $G[t] = (V, E[t])$, consider the execution of a network flood where a node $v \in V$ initiates a flood at time τ by transmitting a message to its neighbors $\mathcal{N}_{G[\tau]}(v)$. The flood continues when v's neighbors transmit to their neighbors so that at time $\tau + 2$, $\mathcal{N}_{G[\tau+1]}(\mathcal{N}_{G[\tau]}(v))$ will receive the message. Continuing the pattern, the s-frontier of the flood, for positive integer s, is given by

$$\mathcal{N}^s_{G[\tau]}(v) := \mathcal{N}_{G[\tau+s-1]}(\mathcal{N}_{G[\tau+s-2]}(\cdots\mathcal{N}_{G[\tau]}(v)))$$

The s-closure of the flood is then the union

$$\mathcal{N}^{s^*}_{G[\tau]}(v) := \mathcal{N}^0_{G[\tau]} \cup \cdots \cup \mathcal{N}^s_{G[\tau]}$$

If for arbitrary initial node v and starting time τ, there exists a positive integer s such that $\mathcal{N}^{s^*}_{G[\tau]}(v) = V$, then we say that $G[t]$ is *floodable*. So far we have assumed that nodes may re-transmit the message multiple times. If we limit the number of re-transmissions to n, and there still exists an s s.t. the analogously defined (n,s)-closure equals V, then we say that $G[t]$ is *n-floodable*. If $|V| \geq k$ and after the removal of an arbitrary set of k nodes from V, $G[t]$ is still n-floodable, then we say that $G[t]$ is (k,n)-*floodable*. Ultimately, we can now state that cooperative nodes will eventually hear all accusations and have the same accusation graph despite up to F Byzantine robots:

Theorem 4 (Eventual Blocklist Consensus). *Let $G[t]$ be the time-varying, (F,n)-floodable network topology of the robot swarm V. If $|\bar{\mathcal{C}}| \leq F$, there $\exists \tau \in \mathbb{Z}^+, \mathcal{A}\forall i \in \mathcal{C}, s \geq \tau$ s.t.*

$\mathcal{A} = \mathcal{A}_i[s]$.

Proof. By definition of (F,n)-floodable, we have that all accusations made by V will eventually reach all of $\mathcal{C}$, since the cooperative robots can ensure eventual delivery of an accusation to all of $\mathcal{C}$ even if up to F Byzantines do not forward accusations. Given that the MRS is closed, the number of possible accusations is finite (bounded by $2|\mathcal{C}| + |\bar{\mathcal{C}}|^2$) and therefore there exists a time τ after which no new accusations are made. As each cooperative robot uses the same deterministic algorithm to compute maximum matchings on the accusation graph, each cooperative robot will eventually compute the same maximum matching and arrive at the same list of robots to block. □

If the assumption that $G[t]$ is (F,n)-floodable does not hold, then some cooperative robot(s) may not receive some of the accusations. If the $R_k[t]$ are not eventually equivalent across all k, it is possible that not all uncooperative robots are blocked (even though globally, enough accusations have been made to satisfy the Hall marriage condition). However, all of $\bar{\mathcal{C}}$ will be blocked by k provided that k's local accusation graph $\mathcal{A}_k[t]$ satisfies the Hall marriage condition, but the matched cooperative robots on the blocklist may not be the same as those on other blocklists.

5.4 Case Studies

We run our experiments on turtlebots simulated in ARGoS [Pinciroli et al., 2012], a multi-physics robot simulator that can efficiently simulate large-scale swarms of robots. The robots are equipped with a radio to transmit to neighboring robots within 4m and have an omnidirectional camera used for nearby target detection and collision avoidance with an observation distance of $\approx$ 0.9m. The robot controller runs at 30Hz.[1]

5.4.1 Target Tracking

Application overview. In swarm target tracking, the goal of the robots is to locate and cooperatively follow a mobile target that has a maximum speed of d. In our experimental

[1] If our paper is accepted, we will release all code needed to run our experiments in ARGoS. All parameters and scenarios can be easily modified via the experiment configuration files.

setup, the target is a robot that has a yellow light – robots within a distance r can see the light and make a direct observation of the target. To enable the entire swarm to track the target, even for those robots that do not directly observe the target, robots broadcast target observation messages containing:

1. the observer's unique ID

2. the time of the observation

3. the observed location of the target

In each timestep, robots sort received observation messages by observation time, and choose the most recent one to transmit to its neighbors. Robots keep track of how many times a given observation message has been transmitted, and stop sharing it after fixed, finite number of times. The purpose of transmitting the same observation message multiple times is to account for the time-varying connectivity with neighboring robots. In addition to the application messages, DBP is used to mitigate the influence of Byzantine robots. Robots delete and do not forward observations messages from blocked observer IDs. Old observation messages are periodically deleted from the local cache.

Controller. For robots that directly observe the target, they compute a heading vector pointing to the target from their current location and move towards the target. Robots that do not directly observe the target rely on received observation messages to compute their heading vector. We denote by $\mathcal{U}_d(c)$ the closed square centered at c with side length $2d$. Given an observation message with time s and observed target location $\tilde{x}$, the implied belief is that the set $\mathcal{U}_{d(t-s)}(\tilde{x})$ contains the target at the current time $t > s$. First, the received observation messages are sorted by time $(s_1, \tilde{x}_1), (s_2, \tilde{x}_2), \ldots$ with $s_1 \geq s_2 \geq \cdots$. To compute the heading vector, robots iteratively take the intersection

$$\mathcal{U}_{d(t-s_1)}(\tilde{x}_1) \cap \mathcal{U}_{d(t-s_2)}(\tilde{x}_2) \cap \cdots$$

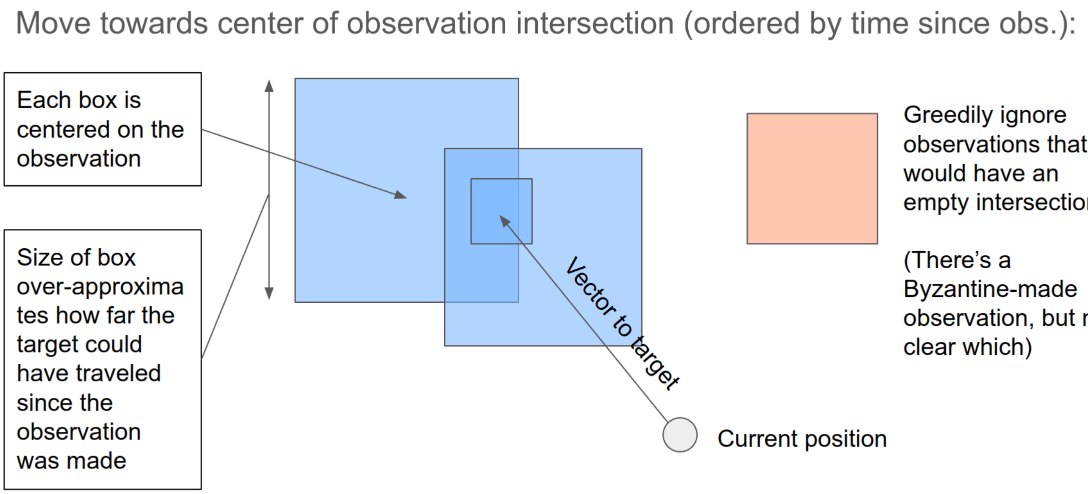
Move towards center of observation intersection (ordered by time since obs.):
Each box is centered on the observation
Size of box over-approximates how far the target could have traveled since the observation was made
Vector to target
Current position
Greedily ignore observations that would have an empty intersection
(There's a Byzantine-made observation, but not clear which)

Figure 5·2: Observation-based target tracking setup for use with DBP. Robots that do not observe the target directly sort received observations by age and compute a bounding box for each observation containing the target based on the elapsed time. Reducing over the bounding boxes with the set intersection operator yields the robot's current belief about the target location. Conflicting observations, those that result in an empty intersection, are dropped, ending the iteration.

If the intersection ever becomes empty while iterating, the offending observation is dropped and the iteration ends. Robots take the center of the intersection to be their believed target location and use it to compute their heading vector. The control procedure is illustrated in Fig. 5·2. Bounding boxes are used instead of circles to simplify the computation of set intersections.

Accusation rules. On receiving a new observation message, robots issue DBP accusations according to four target tracking-specific accusation rules. Given the received observation by robot j of $\tilde{x}$ made at time s, let $\Delta t = t - s$ the elapsed time, $\Delta p_i = \|p_i[t] - \tilde{x}\|$ the distance from i's location $p_i[t]$ to the observed target, and c a constant denoting an upper bound on the speed with which messages can travel through the network (in our experimental setup, 4m/timestep). The first accusation rule is triggered when $r + c\Delta t < \Delta p_i$, as the observation would need to have traveled faster-than-possible through the network.

The second accusation rule is triggered when $\Delta p_i < r - d\Delta t$ and i did not make a direct observation of the target – i missed an observation that it should have made if the received observation was legitimate. The third accusation is rule is triggered when $\Delta p_i > r + d\Delta t$ but a direct observation *was* made by i; in this case the target couldn't possibly have moved fast enough from the received observation location to the place where i observed it presently. Finally, the last accusation rule detects oscillations from a single observer. If i has received an observation from j in the past, it will consider the most recent previous observation from j of $\tilde{x}_{\text{old}}$ at time s_{old}, and will make an accusation of j if $\|\tilde{x} - \tilde{x}_{\text{old}}\| > d(s - s_{\text{old}})$. In this case, j's observations are inherently inconsistent with the maximum rate of change in x.

Experiment setup. We compare DBP-based Byzantine-resilient target tracking with the state-of-the-art W-MSR-based approach. Aside from not needing to know the number of Byzantine robots to tolerate a priori and lower network connectivity requirement, *our approach requires just one non-blocked cooperative robot to observe the moving target*, whereas W-MSR requires $F + 1$ cooperative observers to shift the consensus among the cooperative robots as the target moves. We simulate $|\mathcal{C}| = 200$ and $|\bar{\mathcal{C}}| = 100$ robots to compare tracking performance. Byzantine robots may transmit observation messages and accusations with arbitrary contents. In our scenarios, the behavior of the Byzantine robots is to distribute evenly through the environment and to continuously broadcast false observations – each Byzantine robot picks the location ~ 0.4m away from itself directed away from the origin as the broadcast observation. This Byzantine strategy attempts to lower the network connectivity by causing the cooperative robots to spread out and away from the origin, while simultaneously not violating the speed of network accusation rule.

Experiment result. In Fig. 5·4 we plot the belief that each cooperative robot has about the x-coordinate of the target, summarized using a quantile heatmap. The range of beliefs decreases until approx. $t = 400$, at which point all of the Byzantine robots have been blocked and the execution enters the regime with all Byzantine influence removed. Views

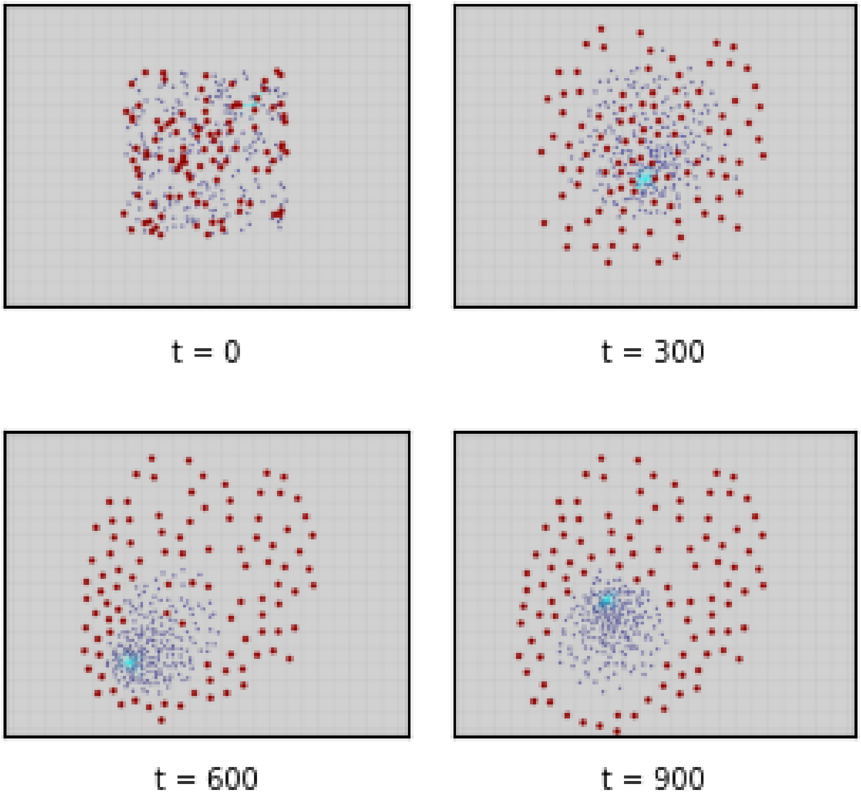

Figure 5·3: View of DBP-based target tracking in ARGoS. Byzantine robots are highlighted with red circles, direct observations of the target are shown in cyan.

of the DBP target tracking experiment in the ARGoS simulator are shown in Fig. 5·3. The baseline W-MSR algorithm requires the resilience parameter F to be picked a priori. If F is chosen too small, the theoretical guarantees of W-MSR are forfeited so the cooperative robots' consensus may be disrupted by the Byzantine robots. Specifically, part of the swarm where the density of Byzantine robots is low may be able to track the target successfully, however cooperative robots with more than F Byzantine neighbors will be affected by the attack. Each cooperative robot affected by the attack will in turn strengthen the attack as their local value nears the attacker's value – this scenario is shown with $F = 15 < |\bar{C}|$ at

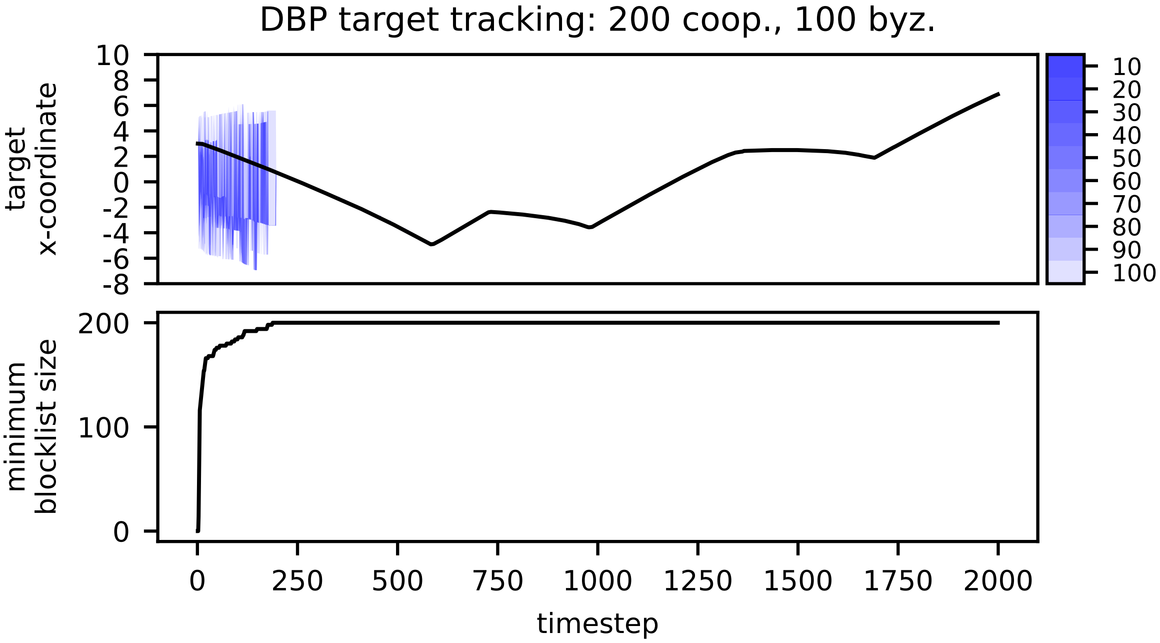

Figure 5·4: DBP-based target tracking performance. At top, the black curve shows the true x-coordinate of the moving target and the shaded blue regions show the range of beliefs as percentiles around the median. At bottom we plot $\min_i R_i^*[t]$, i.e. the minimum blocklist size. At timestep ~ 200, all of the Byzantine robots have been blocked on each cooperative robot, and the cooperative robots track the target with close to no error as the influence of the Byzantines has been removed.

the bottom of Fig. 5·5. However, W-MSR does not scale to large F, since the robots cannot achieve such a high level of network connectivity and also high number of cooperative observers. The large-F regime is shown at top in Fig. 5·5, with $F = 100 = |\bar{C}|$.

5.4.2 Time Synchronization

Application overview. For this task, the robots' objective is to cooperatively synchronize their local clocks to a universal reference clock while moving through the environment. A subset of the robots are designated as *anchors* – these robots periodically make high-precision observations of the reference clock time. As in the target tracking application, the anchors broadcast observation messages containing:

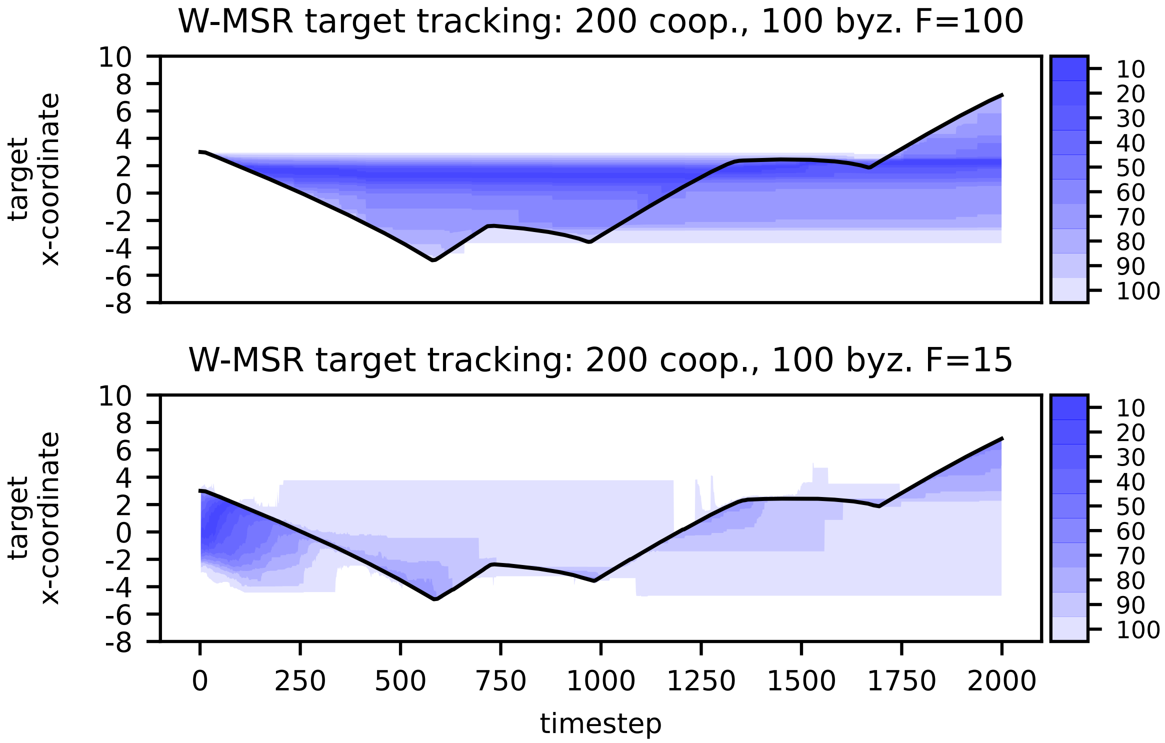

Figure 5·5: W-MSR-based target tracking performance. When the resilience parameter $F = 100$ (top) in order to guarantee safety, the information about the moving target cannot propagate through the cooperative robots due to the high connectivity and simultaneous observer requirements. As a comparison, $F = 15$ (bottom) has no safety guarantee but allows a subset of the robots to track the target successfully. However, the influence of the Byzantines is never removed.

1. the observer's unique ID
2. the observed time

In each timestep, the non-anchor robots sort received observation messages by the observed time in decreasing order and choose the largest value to re-transmit to neighbors, and DBP is used to delete and selectively not forward observation messages from blocked observers.

Controller. On those timesteps when new observation messages are received, non-anchor robots simply update their local clock by setting it to the maximum observed time in their list of observation messages. If a new observation message is not received during a timestep, a non-anchor robot i's local clock is updated by adding a number sampled from the distribution $1+\mu_i+U[-0.05,0.05]$, where $U[a,b]$ is the uniform distribution on $[a,b]$ and μ_i is sampled at the beginning of the simulation from $U[-0.01,0.01]$. This update behavior is intended to simulate a random-walk clock drift when no new observation messages are received.

Accusation rules. Whenever an anchor robot receives a new observation message, it issues an accusation of the origin if the observed time is larger than the anchor's local time. The intuition behind this accusation rule is that the difference between the received observed time and the anchor's local time can only be negative – if the observer is cooperative then the difference should correspond to the number of hops that the observation made on a shortest path to the receiving anchor. If the difference were to be positive, this would imply that the observer's local clock is ahead, violating the assumption that cooperative anchors make high-precision observations of the reference time.

Experiment setup. We compare DBP-based Byzantine-resilient time synchronization with the state-of-the-art W-MSR-based approach. We simulate $|\mathcal{C}|=100$ (50 of which act as anchors, with observation period of 100 timesteps) and $|\bar{\mathcal{C}}|=45$ robots to compare the synchronization performance. Byzantine robots may send arbitrary observation messages, including impersonating anchors. The behavior of the Byzantines in our experiments is to move through the environment just as the cooperative robots do, while broadcast false reference clock observations with the same period as the cooperative anchors. The false observations are the true reference clock value, plus an attack offset of $+1000$ timesteps. This choice of Byzantine adversary attempts to disrupt the time synchronization of the cooperative nodes by forcing the non-anchors to adopt local clock values that are too large – too-low values would be ignored by cooperative robots since each non-anchor always sets their local clock to the maximum observed clock value.

Experiment result. In Fig. 5·6 we plot the time synchronization error (difference be-

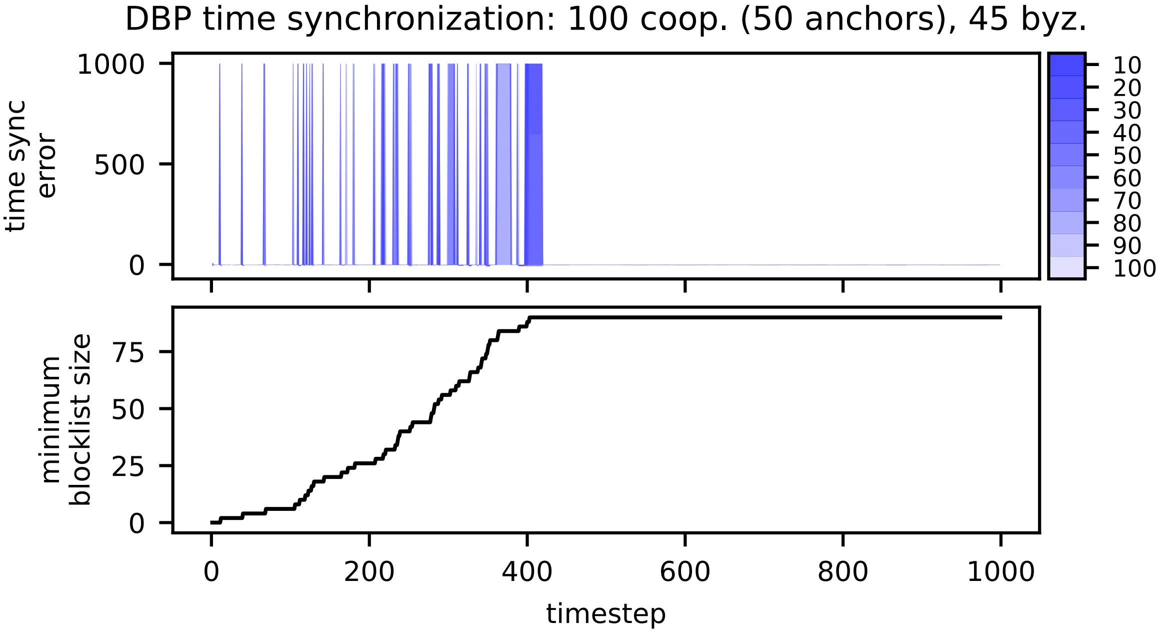

Figure 5·6: DBP-based time synchronization performance. Shown at top is the error between the cooperative robots' local clocks relative to the global reference clock. The error spikes to the attacker offset whenever a Byzantine robot initiates an attack. After timestep ~ 400, all of the Byzantines have been blocked and the tracking error remains nominal.

tween local time and the reference time) of the cooperative non-anchor robots over the course of the simulation. We observe that the Byzantine robots are able to push the synchronization error to the attack offset of $+1000$ timesteps by transmitting the false reference time observations, until timestep ~ 400, at which point each cooperative robot has blocked all of the Byzantine robots and the attacker influence is successfully removed. As with the target tracking case study, the W-MSR baseline requires a choice to be made a priori for the resilience parameter, F. If F is chosen too small, e.g. $F = 10$ shown at bottom in Fig. 5·7, then the Byzantine robots influence the consensus and the cooperative robots have local clock values between the attack offset and the reference time. If F is chosen large enough to be resilient to $|\bar{\mathcal{C}}| = 45$ attackers, then the connectivity and simultaneous observation requirement is too large for the non-anchor robots to update their local clocks from

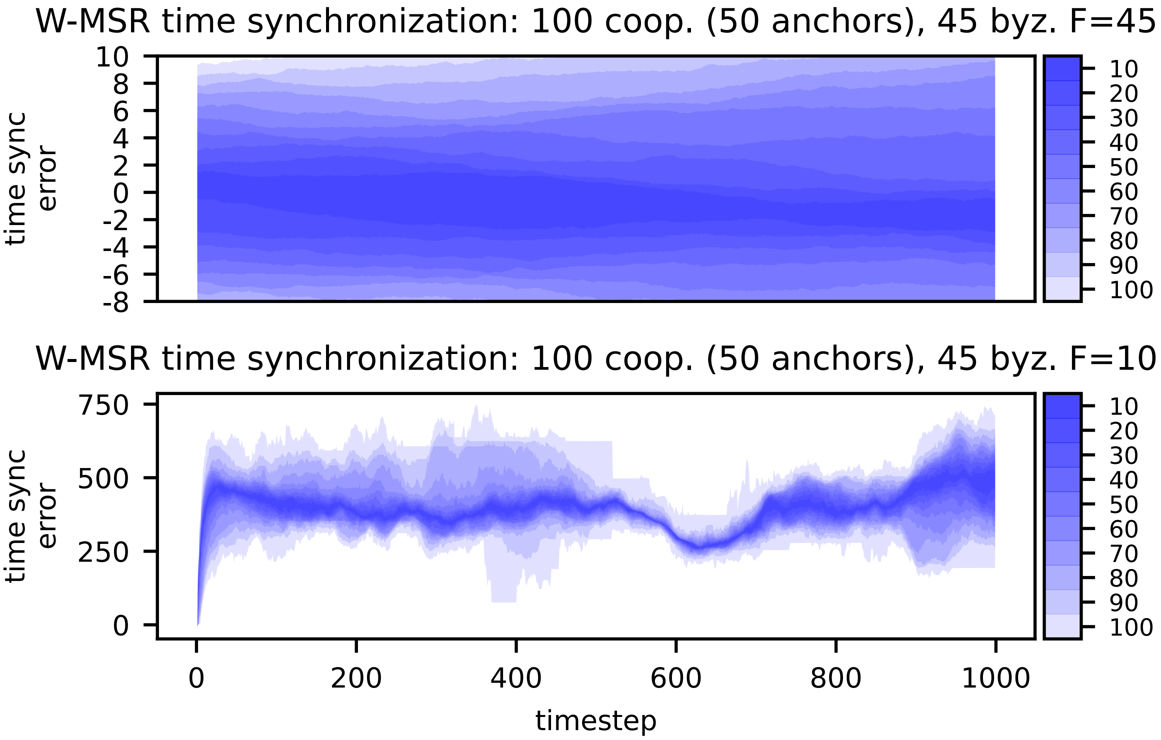

Figure 5·7: W-MSR-based time synchronization performance. As in the target tracking case study, $F = |\bar{\mathcal{C}}| = 45$ guarantees safety, but the associated connectivity requirement prohibits convergence. Conversely, a lower $F = 10$ permits convergence of the consensus at the cost of allowing the Byzantines to adversely perturb the cooperative robots' local clock values.

neighbor's observations. The large-F regime is shown at top in Fig. 5·7 with $F = 45 = |\bar{\mathcal{C}}|$.

5.4.3 Cooperative Localization

Application overview. In the cooperative localization task, robots move in an unknown and/or dynamic environment and use local inter-robot distance measurements to estimate their position within a global coordinate system. To facilitate this task, a subset of the robots operate as anchors, and periodically make high-precision observations of their position (e.g. as static, pre-positioned anchors or mobile robots with GPS). As opposed to the target tracking and time synchronization applications, non-anchor robots also broadcast a localization message containing their localization belief. The localization message contains:

- the sender's unique ID
- the sender's local time
- the sender's believed localization, expressed as bounding box
- an anchor flag, set if and only if the sender is an anchor
    - if the anchor flag is not set, the most recently received anchor localization message

Non-anchor robots initially have no belief about their localization. Once a belief is formed (initially, just the anchors), non-anchors begin to periodically broadcast localization messages to their neighbors. The anchor flag will be set only if the sender is an anchor. Localization messages from non-anchors will be ignored unless the message includes an attached localization message with the anchor flag set.

Controller. On those timesteps when localization messages are received, non-anchor robots sort received localization messages by the time of the underlying anchor message (most recent first), and then use a stable sort to sort by anchor flag (anchor messages first). After sorting, the robot iterates over the received localization messages and takes the intersection of each localization belief, dilated by the transmission range, c, plus the maximum distance a robot can travel per timestep, d. If the intersection ever becomes empty while iterating, the last localization message is dropped and the iteration ends. The resulting intersection is the bounding box that represents the robot's new localization belief. In the next timestep, the robot will transmit its localization belief, bundling the most recent anchor message encountered during the iteration (this may be a direct transmission from an anchor, or an anchor message that arrived as an attachment to a non-anchor's message). The algorithm's operation is illustrated in Fig. 5·8. DBP is used to delete and ignore messages from blocked senders.

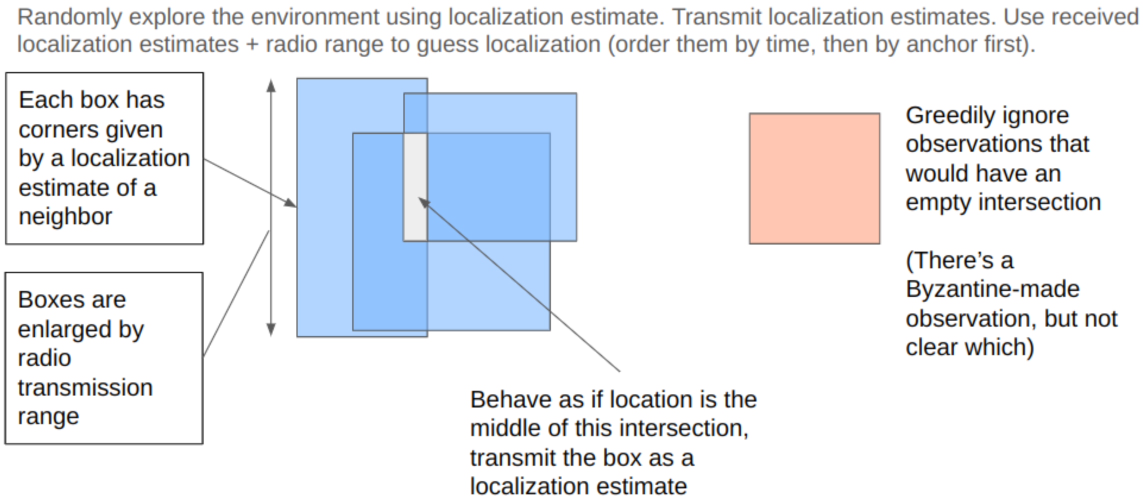
Randomly explore the environment using localization estimate. Transmit localization estimates. Use received localization estimates + radio range to guess localization (order them by time, then by anchor first).
Each box has corners given by a localization estimate of a neighbor
Boxes are enlarged by radio transmission range
Behave as if location is the middle of this intersection, transmit the box as a localization estimate
Greedily ignore observations that would have an empty intersection
(There's a Byzantine-made observation, but not clear which)

Figure 5·8: Observation-based cooperative localization setup for use with DBP. Non-anchor robots estimate their localization based on localization estimates received from their neighbors. Estimates are dilated by the transmission distance, and then reduced with the set intersection operator to compute the localization belief. Localization estimates are ordered by the age of the underlying anchor message, with estimates sent directly from anchors given priority.

Accusation rules. Received localization messages are subjected to two accusation rules. The first rule is applied by anchor robots when receiving localization messages from other anchors, either directly or as attachments to non-anchor localization messages. Given that the other anchor j claims to be at $\tilde{p}_j$ at time s, let $\Delta t = t - s$ the elapsed time and $\Delta x_i = \|\tilde{p}_j - p_i\|$. The receiving anchor i will accuse j if $c\Delta t < \Delta x_i$, or in other words, if the anchor j's localization message has traveled faster-than-possible through the network. The second accusation rule can be issued by all robots, including non-anchors. The second rule asserts that the first rule hold between any received non-anchor localization message and its attached anchor message. These simple accusations could be extended if the robot capabilities were better. For example, if the robots could measure a lower bound on the distance from senders, anchors would be able to issue analogous accusations in situations where localization messages from other anchors should have been received sooner.

Experiment setup. The W-MSR algorithm cannot be chosen as a baseline for this case study, as cooperative localization is not solved via linear consensus problem outside of small-scale settings where each robot can directly observe every other robot in the swarm. We instead demonstrate our approach as a proof-of-concept for Byzantine-resilient cooperative localization. We simulate $|\mathcal{C}| = 120$ (80 of which act as fixed-position anchors) and $|\bar{\mathcal{C}}| = 50$. The Byzantine robots, which attempt to disrupt the localization of the cooperative non-anchors, transmit false anchor localization messages by taking their true position and adding a random attack offset to the x- and y-coordinates sampled uniformly from [-20,20]m. The impact of the false anchor messages on non-anchor robots is to disrupt the iteration over localization messages – since the false anchor localization will likely have an empty intersection with localization messages from nearby cooperative anchors, leading to degraded cooperative localization performance.

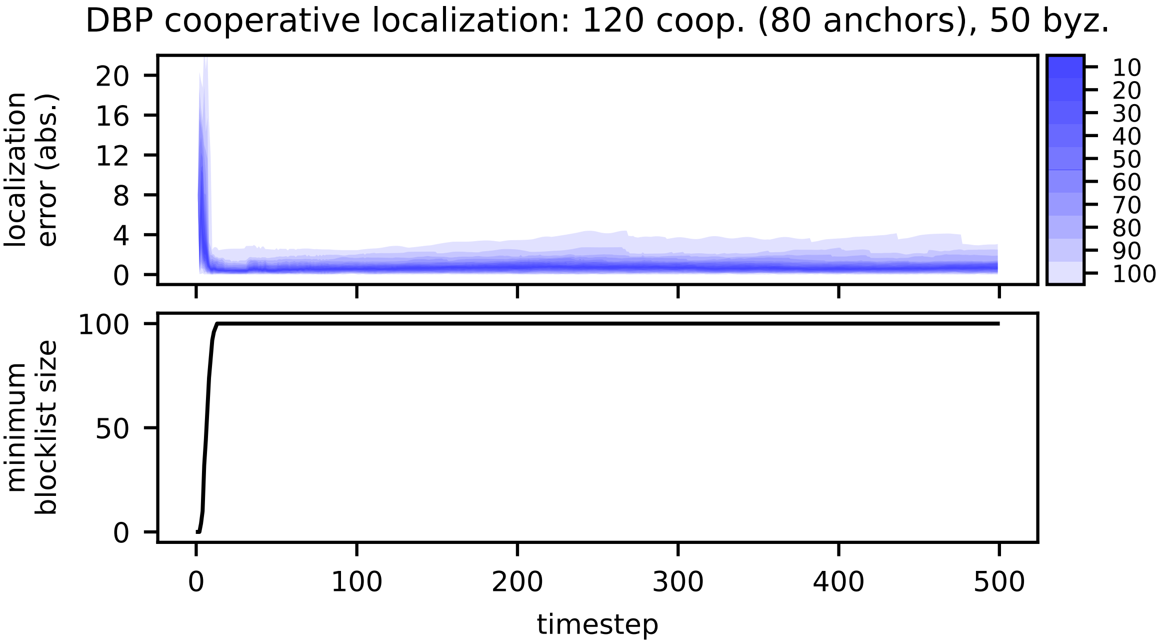

Figure 5·9: DBP-based cooperative localization performance. At top, we plot the absolute error that the cooperative robots have in the estimate of the x-coordinate of their position. At bottom we plot the minimum size of the cooperative robots' blocklists – once all of the Byzantines are blocked the estimation error returns to nominal values as the influence of the Byzantines has been mitigated.

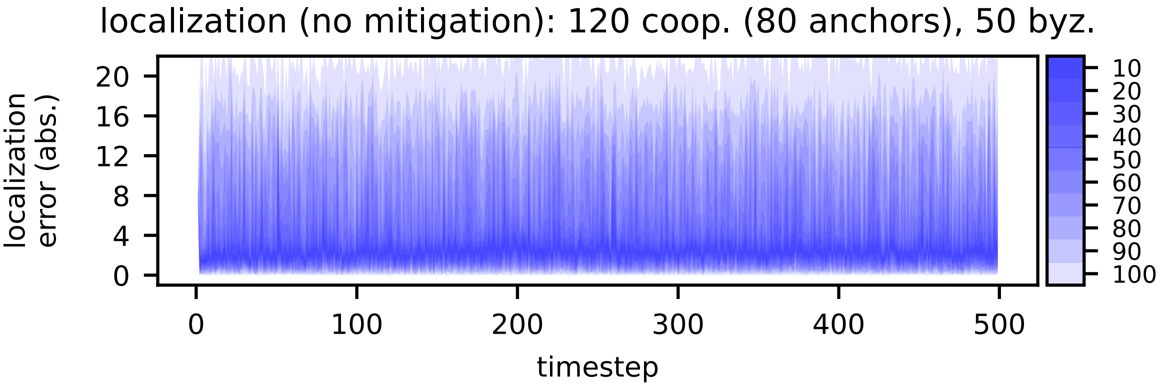

Figure 5·10: To support our claim that DBP is a suitable approach for this task, we show the impact that Byzantine robots can have on cooperative localization – Byzantines can cause the cooperative robots to have arbitrarily large localization errors.

Experiment result. In Fig. 5·9 we plot the absolute error that the cooperative non-anchor robots have in their x-coordinate, i.e. the absolute difference between what they believe their x-coordinate to be and the ground truth. We observe that while initially the cooperative non-anchors may have errors near the attack offset of $\sim$ 20m, the Byzantine robots are rapidly accused and blocked by the cooperative robots. After the Byzantine robots have been blocked, the anchor localization sharing algorithm provides low-error cooperative localization for the non-anchor robots. As a point of comparison, we also simulate the same scenario with DBP disabled, with the absolute x-coordinate localization error shown in Fig. 5·10. As expected, the Byzantine robots significantly disrupt the localization, causing the cooperative non-anchor robots to have consistently high errors up to the attack offset.

5.5 Summary

This chapter has proposed the use of a decentralized blocklist protocol based on inter-robot accusations as a means to provide Byzantine resilience for multi-robot systems. We have shown that as an alternative to the W-MSR algorithm, our approach permits temporary Byzantine influence while accusations are made, but in exchange adapts to Byzantine robots as they are detected, allows for fast information propagation, and can be applied for applications beyond consensus. Based on empirical evidence from swarm target tracking, time synchronization, and localization case studies, our approach is more practical than W-MSR in terms of scalability to large swarms as it does not require each cooperative robot to have $2F+1$ neighbors, nor does it require $F+1$ cooperative observers for information to propagate. In fact, our approach only requires that messages are delivered by network floods in spite of F Byzantine robots, and observations from a single cooperative robot can propagate quickly through the entire swarm. Furthermore, we have shown that our approach can for the first time provide Byzantine resilience for the large-scale decentralized cooperative localization problem.

Chapter 6

Conclusions

In conclusion, this dissertation proposes novel techniques to enhance the security and fault-tolerance of MRSs in uncertain and adversarial environments through inter-robot observations and accusations. First, a fundamental security property for MRSs is proposed, which ensures the detection of forbidden deviations from a desired multi-robot motion plan by the system supervisor. The concept of co-observations is introduced to supplement self-reported motion information and formalized as a method of identifying deviations. An optimal deviation-detecting motion planning problem is formulated to guarantee that all forbidden plan-deviation attacks trigger co-observation-based detection. Second, a horizon-limiting algorithm is proposed to further improve the security and resilience of MRSs against plan deviation attacks by limiting the information available to attackers. Finally, the DBP method is introduced, which allows cooperative robots to identify misbehavior through co-observations and share this information through the network, making it more scalable and effective than existing methods. These techniques will aid in developing secure MRSs for emerging applications in various industries, such as manufacturing, logistics, agriculture, defense, search and rescue, and transportation. In this section, we conclude the dissertation and discuss limitations and present avenues for future consideration.

6.1 Observation Schedules

We introduced our deviation-detecting planner based on inter-robot observations in Chapter 3. Our approach is both complete and makespan-optimal, and provides guaranteed detection of forbidden plan-deviations with a single deviating robot. A promising direction for future work would be to extend the setting to situations where multiple robots deviate simultaneously. Additionally, the cost of having a makespan-optimal plan is a high computation time, which prevents scaling the planner to settings with large numbers of robots. An avenue to improve the scalability of the deviation-detecting planner would be to reduce the planning complexity by making the solution space smaller. For example, by grouping robots in the system into pairs that travel together to trivially ensure that each robot is always observed by a peer. Finally, inter-robot observations are relevant to MRS problems beyond security. In fact, recent efforts in ensuring bounded-time window connectivity for MRS require similar guarantees as provided by deviation-detecting plans. In theory, our work can be applied to improve network guarantees for MRS.

6.2 Announcement Schedules

In Chapter 4 we further studied physical security for centralized MRSs – we described likely compromise scenarios, attacker models, and showed that limiting attacker access to information is able to guarantee prevention of stealthy attackers. However, our approach assumes that compromised robots are not able to collude. If the compromised robots can collude, then they would be able to coordinate their deviations and planned co-observation reports to the supervisor. More work is needed to study the attack planning problem with colluding attackers. Additionally, a topic not considered in this dissertation is how the HoLA verification procedure should influence the motion planner. In other words, when the verification procedure fails for a particular robot, how should that information be leveraged to improve the security of the system. For example, a holistic, hierarchical approach leveraging such information may delay application tasks for robots without monitoring guarantees, or otherwise co-opt those robots to perform dedicated monitoring actions, rather than application tasks.

6.3 Decentralized Blocklist

Chapter 5 represents a significant advancement over the state-of-the-art for Byzantine-resilience in decentralized MRSs. The general-purpose nature of our DBP approach means that we were able to provide for the first time Byzantine-resilience for a cooperative localization application. It is likely that DBP will be used to provide Byzantine-resilience in other settings that previously had no solution. Presently, deriving sound accusation rules requires significant effort and detailed knowledge of the application. Automating accusation rule discovery for novel applications would be useful in accelerating adoption of DBP in complex decentralized MRSs. Similarly, generalizing DBP to situations where accusation rules are not sound, e.g. accusations are only correct with some probability, will allow DBP to be used when no sound accusation rules can be determined. Orthogonal to the study of accusation rule design, another avenue for future work is designing effective motion controllers for use with DBP. Ultimately, accusations are made on the basis of inter-robot observations and those observations only occur if the motion controller encourages cooperative robots to make such observations. Future research could focus on deriving motion controllers that encourage accusations to be made, given the set of accusation rules.

References

(2021). *Industrial Robots and Robot System Safety*, chapter 4. Occupational Safety and Health Administration, United States Department of Labor. Accessed 4 April 2021.

Aditya, U. S., Singh, R., Singh, P. K., and Kalla, A. (2021). A survey on blockchain in robotics: Issues, opportunities, challenges and future directions. *Journal of Network and Computer Applications*, 196:103245.

Agmon, N., Kaminka, G. A., and Kraus, S. (2011a). Multi-robot adversarial patrolling: facing a full-knowledge opponent. *Journal of Artificial Intelligence Research*, 42:887–916.

Agmon, N., Kaminka, G. A., and Kraus, S. (2011b). Multi-robot adversarial patrolling: Facing a full-knowledge opponent. *J. Artif. Int. Res.*, 42(1):887–916.

Agmon, N., Kraus, S., and Kaminka, G. A. (2008a). Multi-robot perimeter patrol in adversarial settings. In *IEEE International Conference on Robotics and Automation*, pages 2339–2345. IEEE.

Agmon, N., Kraus, S., and Kaminka, G. A. (2008b). Multi-robot perimeter patrol in adversarial settings. In *2008 IEEE International Conference on Robotics and Automation*, pages 2339–2345.

Amirian, N. and Shamaghdari, S. (2021). Distributed resilient flocking control of multi-agent systems through event/self-triggered communication. *IET Control Theory & Applications*, 15(4):559–569. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1049/cth2.12061.

Arrichiello, F., Marino, A., and Pierri, F. (2015a). Observer-based decentralized fault detection and isolation strategy for networked multirobot systems. *IEEE Transactions on Control Systems Technology*, 23(4):1465–1476.

Arrichiello, F., Marino, A., and Pierri, F. (2015b). Observer-based decentralized fault detection and isolation strategy for networked multirobot systems. *IEEE Transactions on Control Systems Technology*, 23(4):1465–1476. Publisher: IEEE.

Arrichiello, F., Marino, A., and Pierri, F. (2015c). Observer-based decentralized fault detection and isolation strategy for networked multirobot systems. *IEEE Transactions on Control Systems Technology*, 23(4):1465–1476.

Ashkenazi, Y., Dolev, S., Kamei, S., Ooshita, F., and Wada, K. (2019). Forgive & Forget: Self-Stabilizing Swarms in Spite of Byzantine Robots. In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*, pages 188–194.

Atzmon, D., Stern, R., Felner, A., Wagner, G., Barták, R., and Zhou, N.-F. (2020). Robust Multi-Agent Path Finding and Executing. *Journal of Artificial Intelligence Research*, 67:549–579.

Barer, M., Sharon, G., Stern, R., and Felner, A. (2014a). Suboptimal variants of the conflict-based search algorithm for the multi-agent pathfinding problem. *Frontiers in Artificial Intelligence and Applications*, 263(SoCS):961–962.

Barer, M., Sharon, G., Stern, R., and Felner, A. (2014b). Suboptimal Variants of the Conflict-Based Search Algorithm for the Multi-Agent Pathfinding Problem. *Proceedings of the 7th Annual Symposium on Combinatorial Search.*

Berg, M. Killer robot swarms, an update - POLITICO. `https://www.politico.com/newsletters/digital-future-daily/2023/02/07/killer-robot-swarms-an-update-000816`

Bhattacharya, S., Likhachev, M., and Kumar, V. (2010). Multi-agent path planning with multiple tasks and distance constraints. pages 953–959. IEEE.

Bicchi, A., Danesi, A., Dini, G., La Porta, S., Pallottino, L., Savino, I. M., and Schiavi, R. (2008a). Heterogeneous wireless multirobot system. *IEEE robotics & automation magazine*, 15(1):62–70.

Bicchi, A., Danesi, A., Dini, G., Porta, S. L., Pallottino, L., Savino, I. M., and Schiavi, R. (2008b). Heterogeneous wireless multirobot system. *IEEE Robotics Automation Magazine*, 15(1):62–70.

Bijani, S. and Robertson, D. (2014a). A review of attacks and security approaches in open multi-agent systems. *Artificial Intelligence Review*, 42(4):607–636.

Bijani, S. and Robertson, D. (2014b). A review of attacks and security approaches in open multi-agent systems. *Artificial Intelligence Review*, 42(4):607–636.

Brunner, M., Hofinger, H., Krauss, C., Roblee, C., Schoo, P., and Todt, S. (2010). Infiltrating critical infrastructures with next-generation attacks. `http://publica.fraunhofer.de/documents/N-151330.html`.

Čapek, K. (2020). *RUR*. Standard Ebooks.

Cheng, M., Yin, C., Zhang, J., Nazarian, S., Deshmukh, J., and Bogdan, P. (2021). A general trust framework for multi-agent systems. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*, pages 332–340.

Choi, H., Kate, S., Aafer, Y., Zhang, X., and Xu, D. (2020). Software-based realtime recovery from sensor attacks on robotic vehicles. In *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, pages 349–364.

Choset, H., Lynch, K. M., Hutchinson, S., Kantor, G. A., Burgard, W., Kavraki, L. E., and Thrun, S. (2005a). *Principles of Robot Motion: Theory, Algorithms, and Implementations*. MIT Press.

Choset, H. M., Lynch, K. M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L., Thrun, S., and Arkin, R. C. (2005b). *Principles of robot motion: theory, algorithms, and implementation*. MIT press.

Chu, F., Gailus, S., Liu, L., and Ni, L. (2018). The future of automated ports. *McKinsey & Company*, pages 1–13.

Consulting, N. M. S. Agricultural Robots Market by Type, by Applications, by Component, by Farming Type, and by End Use - Global Opportunity Analysis and Industry Forecast, 2022 – 2030.

De Moura, L. and Bjørner, N. (2008). Z3: An efficient SMT Solver. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 4963 LNCS:337–340.

Dieber, B., Kacianka, S., Rass, S., and Schartner, P. (2016). Application-level security for ROS-based applications. In *International Conference on Intelligent Robots and Systems*, pages 4477–4482. IEEE.

Djouadi, S. M., Melin, A. M., Ferragut, E. M., Laska, J. A., Dong, J., and Drira, A. (2015). Finite energy and bounded actuator attacks on cyber-physical systems. In *2015 European Control Conference (ECC)*, pages 3659–3664. IEEE.

Edmonds, J. (1965). Paths, trees, and flowers. *Canadian Journal of mathematics*, 17:449–467.

Equifax (2017). 2017 cybersecurity incident & important consumer information. `https://www.equifaxsecurity2017.com/`.

Fagiolini, A., Pellinacci, M., Valenti, G., Dini, G., and Bicchi, A. (2008a). Consensus-based distributed intrusion detection for multi-robot systems. In *2008 IEEE International Conference on Robotics and Automation*, pages 120–127. IEEE.

Fagiolini, A., Pellinacci, M., Valenti, G., Dini, G., and Bicchi, A. (2008b). Consensus-based Distributed Intrusion Detection for Multi-Robot Systems. In *2008 IEEE International Conference on Robotics and Automation*, pages 120–127. ISSN: 1050-4729.

Fagiolini, A., Pellinacci, M., Valenti, G., Dini, G., and Bicchi, A. (2008c). Consensus-based distributed intrusion detection for multi-robot systems. In *2008 IEEE International Conference on Robotics and Automation*, pages 120–127.

Fagiolini, A., Valenti, G., Pallottino, L., Dini, G., and Bicchi, A. (2007a). Decentralized intrusion detection for secure cooperative multi-agent systems. In *2007 46th IEEE Conference on Decision and Control*, pages 1553–1558. IEEE.

Fagiolini, A., Valenti, G., Pallottino, L., Dini, G., and Bicchi, A. (2007b). Decentralized intrusion detection for secure cooperative multi-agent systems. In *2007 46th IEEE Conference on Decision and Control*, pages 1553–1558. IEEE.

Fagiolini, A., Valenti, G., Pallottino, L., Dini, G., and Bicchi, A. (2007c). Decentralized intrusion detection for secure cooperative multi-agent systems. In *2007 46th IEEE Conference on Decision and Control*, pages 1553–1558.

Fetch. fetchcore: Cloud robotics platform. `https://fetchrobotics.com/products-technology/fetchcore/`. Accessed: 2018-05-09.

Fisher, A. The Drone That Will Sail Itself Around the World. *Wired*. Section: tags.

Forrest, C. (2017). Robot kills worker on assembly line, raising concerns about human-robot collaboration. `https://www.techrepublic.com/article/robot-kills-worker-\on-assembly-line-raising-concerns-about-human-robot-collaboration/`.

Gabriele, R. and Helmert, M. (2012). Non-Optimal Multi-Agent Pathfinding Is Solved (Since 1984). *Symposium on Combinatorial Search*, (Since 1984):173–174.

Gielis, J., Shankar, A., and Prorok, A. (2022). A Critical Review of Communications in Multi-robot Systems. *Current Robotics Reports*.

Gil, S., Kumar, S., Mazumder, M., Katabi, D., and Rus, D. (2017a). Guaranteeing spoof-resilient multi-robot networks. *Autonomous Robots*, 41(6):1383–1400. Publisher: Springer.

Gil, S., Kumar, S., Mazumder, M., Katabi, D., and Rus, D. (2017b). Guaranteeing spoof-resilient multi-robot networks. *Autonomous Robots*, 41(6):1383–1400.

Gil, S., Kumar, S., Mazumder, M., Katabi, D., and Rus, D. (2017c). Guaranteeing spoof-resilient multi-robot networks. *Autonomous Robots*, 41(6):1383–1400.

Goel, V. and Perlroth, N. (2016). Yahoo says 1 billion user accounts were hacked. `https://www.nytimes.com/2016/12/14/technology/yahoo-hack.html`.

Guerrero-Bonilla, L., Prorok, A., and Kumar, V. (2017). Formations for Resilient Robot Teams. *IEEE Robotics and Automation Letters*, 2(2):841–848. Conference Name: IEEE Robotics and Automation Letters.

Guo, P., Kim, H., Virani, N., Xu, J., Zhu, M., and Liu, P. (2018a). RoboADS: Anomaly detection against sensor and actuator misbehaviors in mobile robots. In *2018 48th Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 574–585. IEEE.

Guo, P., Kim, H., Virani, N., Xu, J., Zhu, M., and Liu, P. (2018b). Roboads: Anomaly detection against sensor and actuator misbehaviors in mobile robots. In *2018 48th Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*, pages 574–585. IEEE.

Gupta, D., Saia, J., and Young, M. (2018). Proof of work without all the work. In *Proceedings of the 19th International Conference on Distributed Computing and Networking*, ICDCN '18, pages 6:1–6:10, New York, NY, USA. ACM.

Gurobi Optimization, LLC (2018). Gurobi optimizer reference manual.

Hönig, W. (2021). libmultirobotplanning.

Hönig, W., Kiesel, S., Tinka, A., Durham, J. W., and Ayanian, N. (2019). Persistent and robust execution of MAPF schedules in warehouses. *IEEE Robotics and Automation Letters*, 4(2):1125–1131.

Honig, W., Kiesel, S., Tinka, A., Durham, J. W., and Ayanian, N. (2019). Persistent and Robust Execution of MAPF Schedules in Warehouses. *IEEE Robotics and Automation Letters*, 4(2):1125–1131.

IEEE Spectrum. Three engineers, hundreds of robots, one warehouse. `https://www.spectrum.ieee.org/robotics/robotics-software/three-engineers-hundreds-of-robots-` Accessed: 2018-04-02.

Ivanova, M. and Surynek, P. (2014). Adversarial Cooperative Path-Finding: Complexity and Algorithms. *Proceedings - International Conference on Tools with Artificial Intelligence, ICTAI*, 2014-Decem:75–82.

Javaid, A. Y., Sun, W., Devabhaktuni, V. K., and Alam, M. (2012). Cyber security threat analysis and modeling of an unmanned aerial vehicle system. In *2012 IEEE Conference on Technologies for Homeland Security (HST)*, pages 585–590.

Ji, M. and Egerstedt, M. (2007). Distributed coordination control of multiagent systems while preserving connectedness. *IEEE Transactions on Robotics*, 23(4):693–703.

Jones, M. (2018). Breach! walmart exposed personal data of 1.3 million u.s. shoppers. `https://www.komando.com/happening-now/446247/breach-walmart-exposed-personal-dat`

Kim, J. M., Choi, J. S., and Lee, B. H. (2008a). Multi-agent coordinated motion planning for monitoring and controlling the observed space in a security zone. *IFAC Proceedings Volumes*, 41(2):1679–1684.

Kim, J. M., Choi, J. S., and Lee, B. H. (2008b). Multi-agent coordinated motion planning for monitoring and controlling the observed space in a security zone. *IFAC Proceedings Volumes*, 41(2):1679–1684. 17th IFAC World Congress.

Koetsier, J. Health Via Drone: Zipline Now Delivering Medicine Via Fixed-Wing Drones In North Carolina. `https://www.forbes.com/sites/johnkoetsier/2022/06/28/health-via-drone-zipline-now-delivering-medicine-via-fixed-wing-drones-in-north-` Section: Consumer Tech.

Kouvaros, P., Lomuscio, A., and Pirovano, E. (2018). Symbolic synthesis of fault-tolerance ratios in parameterised multi-agent systems. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 324–330. International Joint Conferences on Artificial Intelligence Organization.

Krebs, B. (2013). Adobe breach impacted at least 38 million users. `https://krebsonsecurity.com/2013/10/adobe-breach-impacted-at-least-38-million-users/`.

Krebs, B. (2014). The target breach, by the numbers. `https://krebsonsecurity.com/2014/05/the-target-breach-by-the-numbers/`.

Kupferman, O. and Y. Vardi, M. (2001). Model checking of safety properties. *Formal Methods in System Design*, 19(3):291–314.

LaValle, S. M. (2006a). *Planning algorithms*. Cambridge university press.

LaValle, S. M. (2006b). *Planning Algorithms*. Cambridge University Press, New York, NY, USA.

LeBlanc, H. J., Zhang, H., Koutsoukos, X., and Sundaram, S. (2013). Resilient Asymptotic Consensus in Robust Networks. *IEEE Journal on Selected Areas in Communications*, 31(4):766–781. Conference Name: IEEE Journal on Selected Areas in Communications.

Lee, C., Lawry, J., and Winfield, A. (2017). Combining Opinion Pooling and Evidential Updating for Multi-Agent Consensus. Technical report.

Li, J., Tinka, A., Kiesel, S., Durham, J. W., Kumar, T. K. S., and Koenig, S. (2021). Lifelong Multi-Agent Path Finding in Large-Scale Warehouses. *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*.

Liu, J., Zhou, L., Tokekar, P., and Williams, R. K. (2021a). Distributed Resilient Submodular Action Selection in Adversarial Environments. *IEEE Robotics and Automation Letters*, 6(3):5832–5839. arXiv: 2105.07305.

Liu, J., Zhou, L., Tokekar, P., and Williams, R. K. (2021b). Distributed resilient submodular action selection in adversarial environments. *IEEE Robotics and Automation Letters*, 6(3):5832–5839.

Luo, W., Chakraborty, N., and Sycara, K. (2016). Distributed dynamic priority assignment and motion planning for multiple mobile robots with kinodynamic constraints. pages 148–154.

Ma, H., Wagner, G., Felner, A., Li, J., Kumar, T. K. S., and Koenig, S. (2018). Multi-Agent Path Finding with Deadlines. (July).

Mallmann-Trenn, F., Cavorsi, M., and Gil, S. (2021). Crowd Vetting: Rejecting Adversaries via Collaboration With Application to Multirobot Flocking. *IEEE Transactions on Robotics*, pages 1–20. Conference Name: IEEE Transactions on Robotics.

Mastellone, S., Stipanović, D. M., Graunke, C. R., Intlekofer, K. A., and Spong, M. W. (2008). Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments. *The International Journal of Robotics Research*, 27(1):107–126.

Mersetzky, M. Voliro drones to inspect power plants in the USA. https://www.s-ge.com/en/article/news/20223-robotics-constellation-clearsight-uses-voliro-drones.

Mitra, A., Richards, J. A., Bagchi, S., and Sundaram, S. (2019). Resilient distributed state estimation with mobile agents: overcoming Byzantine adversaries, communication losses, and intermittent measurements. *Autonomous Robots*, 43(3):743–768.

Morante, S., Victores, J. G., and Balaguer, C. (2015a). Cryptobotics: Why robots need cyber safety. *Frontiers in Robotics and AI*, 2:23.

Morante, S., Victores, J. G., and Balaguer, C. (2015b). Cryptobotics: why robots need cyber safety. https://www.frontiersin.org/articles/10.3389/frobt.2015.00023/full.

Pacheco, A., Strobel, V., and Dorigo, M. (2020). A blockchain-controlled physical robot swarm communicating via an ad-hoc network. In *International Conference on Swarm Intelligence*, pages 3–15. Springer.

Panagou, D., Stipanovic, D. M., and Voulgaris, P. G. (2013). Multi-objective control for multi-agent systems using lyapunov-like barrier functions. In *Decision and Control (CDC), 2013 IEEE 52nd Annual Conference on*, pages 1478–1483. IEEE.

Pierson, A. and Schwager, M. (2016). Adaptive Inter-Robot Trust for Robust Multi-Robot Sensor Coverage. In Inaba, M. and Corke, P., editors, *Robotics Research*, volume 114, pages 167–183. Springer International Publishing, Cham. Series Title: Springer Tracts in Advanced Robotics.

Pinciroli, C., Trianni, V., O'Grady, R., Pini, G., Brutschy, A., Brambilla, M., Mathews, N., Ferrante, E., Di Caro, G., Ducatelle, F., Birattari, M., Gambardella, L. M., and Dorigo, M. (2012). ARGoS: a modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intelligence*, 6(4):271–295.

Portugal, D., Pereira, S., and Couceiro, M. S. (2017). The role of security in human-robot shared environments: A case study in ROS-based surveillance robots. In *2017 26th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN)*, pages 981–986. IEEE.

Quarta, D., Pogliani, M., Polino, M., Maggi, F., Zanchettin, A. M., and Zanero, S. (2017a). An Experimental Security Analysis of an Industrial Robot Controller. *2017 IEEE Symposium on Security and Privacy (SP)*, pages 268–286.

Quarta, D., Pogliani, M., Polino, M., Maggi, F., Zanchettin, A. M., and Zanero, S. (2017b). An experimental security analysis of an industrial robot controller. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 268–286. IEEE.

Renganathan, V. and Summers, T. (2017a). Spoof resilient coordination for distributed multi-robot systems. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 135–141. IEEE.

Renganathan, V. and Summers, T. (2017b). Spoof resilient coordination for distributed multi-robot systems. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 135–141. IEEE.

Renganathan, V. and Summers, T. (2017c). Spoof resilient coordination for distributed multi-robot systems. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 135–141.

Rivera, S., Lagraa, S., Nita-Rotaru, C., Becker, S., and State, R. (2019). ROS-defender: SDN-based security policy enforcement for robotic applications. In *2019 IEEE Security and Privacy Workshops (SPW)*, pages 114–119. IEEE.

Saha, I., Ramaithitima, R., Kumar, V., Pappas, G. J., and Seshia, S. A. (2014). Automated composition of motion primitives for multi-robot systems from safe ltl specifications. In *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1525–1532.

Saldaña, D., Prorok, A., Sundaram, S., Campos, M. F. M., and Kumar, V. (2017). Resilient consensus for time-varying networks of dynamic agents. In *2017 American Control Conference (ACC)*, pages 252–258. ISSN: 2378-5861.

Salem, M. B., Hershkop, S., and Stolfo, S. J. (2008). *A Survey of Insider Attack Detection Research*, pages 69–90. Springer US, Boston, MA.

Sartoretti, G., Kerr, J., Shi, Y., Wagner, G., Kumar, T. K. S., Koenig, S., and Choset, H. PRIMAL: Pathfinding via Reinforcement and Imitation Multi-Agent Learning.

Saulnier, K., Saldaña, D., Prorok, A., Pappas, G. J., and Kumar, V. (2017). Resilient Flocking for Mobile Robot Teams. *IEEE Robotics and Automation Letters*, 2(2):1039–1046. Conference Name: IEEE Robotics and Automation Letters.

Schedl, D. C., Kurmi, I., and Bimber, O. (2021). An autonomous drone for search and rescue in forests using airborne optical sectioning. *Science Robotics*, 6(55):eabg1188.

Sharon, G., Stern, R., Felner, A., and Sturtevant, N. Conflict-Based Search For Optimal Multi-Agent Path Finding. pages 563–569.

Shoukry, Y., Mishra, S., Luo, Z., and Diggavi, S. (2018a). Sybil attack resilient traffic networks: A physics-based trust propagation approach. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 43–54. IEEE.

Shoukry, Y., Mishra, S., Luo, Z., and Diggavi, S. (2018b). Sybil attack resilient traffic networks: A physics-based trust propagation approach. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*, pages 43–54. IEEE.

Shoukry, Y., Mishra, S., Luo, Z., and Diggavi, S. (2018c). Sybil attack resilient traffic networks: A physics-based trust propagation approach. In *Proceedings of the 9th ACM/IEEE International Conference on Cyber-Physical Systems*, ICCPS '18, pages 43–54, Piscataway, NJ, USA. IEEE Press.

Stern, R., Sturtevant, N. R., Felner, A., Koenig, S., Ma, H., Walker, T. T., Li, J., Atzmon, D., Cohen, L., Kumar, T. K. S., Barták, R., and Boyarski, E. (2019). Multi-Agent Pathfinding: Definitions, Variants, and Benchmarks. In *Twelfth Annual Symposium on Combinatorial Search*.

Strobel, V., Castelló Ferrer, E., and Dorigo, M. (2018). Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems*, AAMAS '18, page 541–549, Richland, SC. International Foundation for Autonomous Agents and Multiagent Systems.

Turpin, M., Michael, N., and Kumar, V. (2014). CAPT: Concurrent assignment and planning of trajectories for multiple robots. 33(1):98–112.

Ulusoy, A., Smith, S. L., Ding, X. C., and Belta, C. (2012a). Robust multi-robot optimal path planning with temporal logic constraints. In *2012 IEEE International Conference on Robotics and Automation*, pages 4693–4698. IEEE.

Ulusoy, A., Smith, S. L., Ding, X. C., and Belta, C. (2012b). Robust multi-robot optimal path planning with temporal logic constraints. In *2012 IEEE International Conference on Robotics and Automation*, pages 4693–4698.

Ulusoy, A., Smith, S. L., Ding, X. C., Belta, C., and Rus, D. (2011a). Optimal multi-robot path planning with temporal logic constraints. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3087–3092. IEEE.

Ulusoy, A., Smith, S. L., Ding, X. C., Belta, C., and Rus, D. (2011b). Optimal multi-robot path planning with temporal logic constraints. In *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 3087–3092.

Vasic, M. and Billard, A. (2013). Safety issues in human-robot interactions. In *2013 ieee international conference on robotics and automation*, pages 197–204. IEEE.

Verma, J. K. and Ranga, V. (2021). Multi-Robot Coordination Analysis, Taxonomy, Challenges and Future Scope. *Journal of Intelligent & Robotic Systems*, 102(1):10.

Wang, K.-H. C. and Botea, A. (2009). Tractable multi-agent path planning on grid maps. In *Proceedings of the 21st International Jont Conference on Artifical Intelligence*, IJCAI'09, pages 1870–1875, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Wang, K.-H. C. and Botea, A. (2011a). A Scalable Multi-Agent Path Planning Algorithm with Tractability and Completenss Guarantees. *JAIR - Journal of Artificial Intelligence Research*, 42:55–90.

Wang, K.-H. C. and Botea, A. (2011b). Mapp: a scalable multi-agent path planning algorithm with tractability and completeness guarantees. *Journal of Artificial Intelligence Research*, 42:55–90.

Wang, K.-H. C., Botea, A., et al. (2009). Tractable multi-agent path planning on grid maps. In *IJCAI*, volume 9, pages 1870–1875. Pasadena, California.

Wardega, K., Tron, R., and Li, W. (2019a). Masquerade Attack Detection Through Observation Planning for Multi-Robot Systems. In *The 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Wardega, K., Tron, R., and Li, W. (2019b). Resilience of Multi-Robot Systems to Physical Masquerade Attacks. In *IEEE Workshop on the Internet of Safe Things (SafeThings)*.

Wardega, K., von Hippel, M., Tron, R., Nita-Rotaru, C., and Li, W. (2023a). Byzantine Resilience at Swarm Scale: A Decentralized Blocklist from Inter-robot Accusations. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Wardega, K., von Hippel, M., Tron, R., Nita-Rotaru, C., and Li, W. (2023b). HoLA Robots: Mitigating Plan-Deviation Attacks in Multi-Robot Systems with Co-Observations and Horizon-Limiting Announcements. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Yaacoub, J.-P. A., Noura, H. N., Salman, O., and Chehab, A. (2021). Robotics cyber security: Vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, pages 1–44.

Yaacoub, J.-P. A., Noura, H. N., Salman, O., and Chehab, A. (2022). Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security*, 21(1):115–158.

Yang, F. and Chakraborty, N. (2019). Multirobot simultaneous path planning and task assignment on graphs with stochastic costs. In *2019 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*, pages 86–88. IEEE.

Yang, H., Staroswiecki, M., Jiang, B., and Liu, J. (2011a). Fault tolerant cooperative control for a class of nonlinear multi-agent systems. *Systems & control letters*, 60(4):271–277.

Yang, H., Staroswiecki, M., Jiang, B., and Liu, J. (2011b). Fault tolerant cooperative control for a class of nonlinear multi-agent systems. *Systems & Control Letters*, 60(4):271 – 277.

Yang, Z. and Tron, R. (2020). Multi-agent path planning under observation schedule constraints.

Yu, J. (2016). Intractability of optimal multirobot path planning on planar graphs. *IEEE Robotics and Automation Letters*, 1(1):33–40.

Yu, J. and LaValle, S. M. (2013). Structure and intractability of optimal multi-robot path planning on graphs. In *Twenty-Seventh AAAI Conference on Artificial Intelligence*.

Yu, J. and LaValle, S. M. (2016). Optimal multirobot path planning on graphs: Complete algorithms and effective heuristics. 32(5):1163–1177.

Zhou, L. and Tokekar, P. (2021). Multi-Robot Coordination and Planning in Uncertain and Adversarial Environments. *Current Robotics Reports*, 2(2):147–157. arXiv: 2105.00389.

Zhou, L., Tzoumas, V., Pappas, G. J., and Tokekar, P. (2020). Distributed Attack-Robust Submodular Maximization for Multi-Robot Planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2479–2485. ISSN: 2577-087X.

CURRICULUM VITAE

Kacper Wardega

Education

2017–2023 **Ph.D.** . . . . Boston University Electrical & Computer Engineering Dept.
Computer Engineering
Dependable Computing Laboratory – Advised by Prof. Wenchao Li

2013–2016 **B.Sc.** (cum laude with honors) . . . . The Ohio State University College of Engineering
Major in Computer Science & Engineering

2013–2016 **B.Sc.** (cum laude) . . . . The Ohio State University College of Arts and Sciences
Major in Mathematics, Minor in German
Specialization in Theoretical Mathematics

Publications

Huang, C., **Wardega, K.**, Li, W., and Zhu, Q. (2019). Exploring Weakly-hard Paradigm for Networked Systems. In *The 1st Workshop on Design Automation for CPS and IoT (DESTION)*.

Kiourti, P., **Wardega, K.**, Jha, S., and Li, W. (2020). Evaluation of Backdoor Attacks on Deep Reinforcement Learning. In *Design Automation Conference (DAC)*.

Wardega, K. and Li, W. (2020). Application-Aware Scheduling of Networked Applications over the Low-Power Wireless Bus. In *Design, Automation and Test in Europe (DATE)*.

Wardega, K., Li, W., Kim, H., Wu, Y., Jia, Z., and Hu, J. (2022). Opportunistic Communication with Latency Guarantees for Intermittently-Powered Devices. In *Design, Automation and Test in Europe (DATE)*.

Wardega, K., Tron, R., and Li, W. (2019a). Masquerade Attack Detection Through Observation Planning for Multi-Robot Systems. In *The 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*.

Wardega, K., Tron, R., and Li, W. (2019b). Resilience of Multi-Robot Systems to Physical Masquerade Attacks. In *IEEE Workshop on the Internet of Safe Things (SafeThings)*.

Wardega, K., von Hippel, M., Tron, R., Nita-Rotaru, C., and Li, W. (2023 (to appear)a). Byzantine Resilience at Swarm Scale: A Decentralized Blocklist from Inter-robot Accusations. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*

Wardega, K., von Hippel, M., Tron, R., Nita-Rotaru, C., and Li, W. (2023 (to appear)b). HoLA Robots: Mitigating Plan-Deviation Attacks in Multi-Robot Systems with Co-Observations and Horizon-Limiting Announcements. In *The 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS).*

Zhu, Q., Li, W., Kim, H., Xiang, Y., **Wardega, K.**, Wang, Z., Wang, Y., Liang, H., Huang, C., Fan, J., and Choi, H. (2020). Know the Unknowns: Addressing Disturbances and Uncertainties in Autonomous Systems. In *International Conference on Computer Aided Design (ICCAD).*

Work Experience

Summer 2021 **Applied Scientist Intern** . . . . Amazon Robotics
Conducted research on the performance of fully-automated robotic floors. Project culminated in a whitepaper, all-hands talk, and software artifacts.

Honors

2021	Graduate Teaching Assistant of the Year Award	BU ECE Dept.
2019	Travel Grant IEEE S&P 2019	IEEE S&P 2019 Award Committee
2019	Travel Grant CPS-IoTWeek 2019	CPS-IoTWeek 2019 Award Committee
2017	Hariri Graduate Student Fellowship	BU Hariri Institute for Computing

Core Graduate Coursework

Fall 2017	EC504: Advanced Data Structures
Fall 2017	EC700: Computer Aided Verification & Synthesis
Spring 2018	EC700: Advanced Computer Systems
Fall 2018	EC505: Stochastic Processes
Fall 2019	EC541: Computer Communication Networks
Spring 2021	EC724: Advanced Optimization Theory & Methods

Academic Service

Peer Reviewing . . . .

1. Design Automation Conference (DAC), 2018, 2019, 2020.
2. Design, Automation and Test in Europe (DATE), 2020, 2021, 2022.
3. Dependable Systems and Networks (DSN), 2021, 2022.
4. International Conference on Embedded Software (EMSOFT), 2018.
5. ACM International Conference on Hybrid Systems: Computation and Control (HSCC), 2020.
6. International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS), 2020, 2023.
7. International Conference On Computer Aided Design (ICCAD), 2018, 2021, 2022.
8. ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec), 2022.
9. Asia and South Pacific Design Automation Conference (ASP-DAC) 2023.
10. IEEE International Conference on Robotics and Automation (ICRA) 2023.

FA 2019, SP 2020 Teaching Assistant . . . . EC330: Applied Algorithms for Engineers
Undergraduate course on algorithms. Topics covered include the general concept of algorithms, efficiency and run-time of algorithms, graph algorithms, priority queues, search trees, various approaches to design of algorithms and data structures, together with their applications to numerical and non-numerical problems.

FA 2019-2022 Grading Assistant . . . . EC545: Cyber-Physical Systems
Graduate course on building high-assurance systems with real-time and concurrent behaviors. Topics covered include modeling of dynamical behavior and the design, specification, and analysis of CPS.

SU 2018-2019 Mentoring . . . . BU RISE
Mentored high school students during six-week summer research internships.