Protecting the Satellite Ecosystem

Rachel McAmis

A dissertation
submitted in partial fulfillment of the
requirements for the degree of

Doctor of Philosophy

University of Washington

2026

Reading Committee:
Tadayoshi Kohno, Chair
David Kohlbrenner
Samuel Mergendahl

Program Authorized to Offer Degree:
Paul G. Allen School of Computer Science & Engineering

University of Washington

Abstract

Protecting the Satellite Ecosystem

Rachel McAmis

Chair of the Supervisory Committee:
Tadayoshi Kohno
Georgetown University

Satellites are becoming increasingly intertwined in our daily lives, whether we are aware of it or not. As the importance and prevalence of satellites continues to increase, so do the implications on the privacy, security, and safety of humans on Earth and in space. These risks are not limited to a particular type of satellite nor a particular adversary. They span the entire satellite ecosystem. For example, state adversaries may seek to damage satellites launched by a competing country for financial gain, or to destroy a portion of satellite-facilitated critical infrastructure as part of cyber warfare. Or, even without a security compromise, individuals could purchase increasingly affordable commercial satellite images for interpersonal surveillance.

This dissertation identifies and addresses security and privacy challenges across the satellite ecosystem that have previously received little attention. First, this work surveys the possible threats of commercial imaging satellites to individual privacy. Commercial imaging satellites pose both possible benefits and privacy harms, and this work explores these tradeoffs through a survey of 99 participants, followed by a discussion of possible solutions that still maintain utility benefits. Second, this work looks at the importance of smaller satellites and argues why they must still practice good security. Because of the cyberphysical effects of satellites, one insecure satellite has the capacity to damage other satellites in the ecosystem. This interconnectedness necessitates careful analysis of security

and other possible harms among all satellite operators of every satellite in orbit. This work puts forth possible threats posed by university smallsats, along with barriers to building them securely through a combination of in-depth interviews and open-source code analysis. Lastly, this work focuses on emerging threats to more complex satellites or fleets of satellites. This work identifies the risks of a compromised peripheral on a large complex satellite. This work demonstrates through simulation that current resiliency techniques do not handle the case when a satellite peripheral is malicious rather than just faulty. Then, the final portion of this work proposes and experimentally tests a novel technique that identifies and takes action on malicious peripherals. Together, this work makes steps toward a safer satellite ecosystem that benefits those on the ground.

Table of Contents

List of Figures

List of Tables

Acknowledgments

From my first quarter of undergrad, I had my sights set on working in the UW Security and Privacy Lab. Little did I know that I would work there for almost seven years, from my sophomore year of undergrad to my final year of grad school. Through all of these years, I had the privilege of working with Yoshi Kohno, who went on to be my PhD advisor. Yoshi inspires me in many ways. He gave me the freedom to explore while still providing structure to support my development as a researcher. Most importantly, Yoshi always made time for me even before he officially became my advisor. Yoshi inspired and supported unconventional research; he even encouraged me to spend weeks of my time on studying to receive a commercial (drone) pilot license and an amateur radio license. He also encouraged me to join the UW satellite club that took many hours away from my research short-term but ultimately made me stronger technically. With his guidance, we challenged norms in research approaches.

Thank you to my committee members from quals to the final exam: Mia Bennett, Jessica Beyer, David Kohlbrenner, Sam Mergendahl, and Saadia Pekkanen. Your guidance helped focus my dissertation goals. My collaborators have all added a unique perspective to the work in this dissertation: Mia Bennett, Gregor Haas, David Kohlbrenner, Yoshi Kohno, Sam Mergendahl, Mattea Sim, Rick Skowyra, and Connor Willison.

For the past couple years I have also been incredibly lucky to do research at MIT Lincoln Laboratory. At Lincoln Lab, Sam Mergendahl provided weekly, in-depth guidance and satellite expertise. He helped me maintain high standards every step of the way, from project implementation to presenting. I have been lucky enough to build my systems atop top-notch Lincoln Lab code/infrastructure. I have also been lucky to collaborate with other

Lincoln Lab engineers, including Connor Willison, who spent time out of his already busy schedule to help debug and provide aerospace expertise. And thank you to Rick Skowyra and Roger Khazan for their program leadership.

Thank you to the security lab members for a fun and supportive work environment. Past and present members of the lab that I have had the privilege of overlapping with include: Grace Brigham, Maddie Burbage, Ian Chang, Kaiming Cheng, Katarina Cheng, Inyoung Cheong, Aarushi Dubey, Yael Eiger, Michael Flanders, Kshitish Ghate, Gregor Haas, Rachel Hong, Alexandra Michael, Kentrell Owens, Miranda Wei, Henry Wong, Tina Yeung, and Eric Zeng. Gregor Haas: thank you for our many fascinating research discussions and for joining me on a really fun research project.

The staff at UW help abstract away many logistical challenges for us students. Thank you to Elise and Joe for handling myriad tasks that arose.

Thank you to the employees at Apex Space for letting me talk at them about security as much as I wanted. And in particular, Nick Gutierrez for mentorship and touring me through my first factory cleanroom.

Starting a PhD would not have been possible without the many helpful mentors and advisors at the University of Washington. Elise Dorough and Maggie Morris: you both helped me believe in myself as someone who could be capable of doing research, and showed me the path to get there. Thank you to my first research mentors: Lee Organick for being the first person who introduced me to what research and a PhD even was, Peter Ney for introducing me to the field of security research, and Eric Zeng for a meticulous approach to mentorship and research that was immensely helpful to my growth as a researcher. David Kohlbrenner, Franzi Roesner, and Nirvan Tiyagi: for helping me develop as a well-rounded researcher through leadership in the lab, whether it be hearing about their work in lab meetings, working alongside them, taking their classes, or getting feedback.

There are so many educators and coaches that I am grateful for throughout my academic

journey. Thank you to Samuel Ring for being an incredible high school running coach that I can still talk to today about both athletic and career advice. You helped me believe in myself as an athlete and showed me how consistent, cumulative effort can lead to long-term success. Not so dissimilar from a PhD! And thank you to Ms. Pinckney for two amazing elementary school years and making learning fun.

I am thankful to have supportive family and friends: Elizabeth and Wayne McAmis for giving me the support and freedom to allow me to focus on things I cared about most, Hunter McAmis for being a great brother, Prithvi Ranjan for being my career and life coach throughout my time at UW, Paul and Christine Carr for encouragement and teaching me how to ask good questions, Julie Umbreit for fostering my writing from a young age, Arthur Coburn for inspiring me to pursue my passions, all of my extended family who enthusiastically listened to me talk about research, Nayha Auradkar for our friendship and being my first coding partner, Taylor Ka for being a great roommate and academic inspiration, Andrea Dang for being up for anything, Thanhvan Nguyen for her supportive optimism, Sophie Baye for being a dependable running buddy, and Kieran Chen-Johnson for laughter even on opposite coasts.

Thank you to the Chickasaw Nation for financially supporting my education during my undergraduate career and beyond.

I am grateful to all sources of funding. My graduate career and dissertation work were supported in part by the University of Washington Tech Policy Lab, National Science Foundation under award CNS-2207019, gifts from Google, Qualcomm, and Woven Planet, and MIT Lincoln Laboratory through internship work (projects supported by the Department of the Air Force under Air Force Contract No. FA8702-15-D- 0001 or FA8702-25-D-B002).

Chapter 1
Introduction

Satellites are becoming increasingly intertwined in our daily lives, whether we are aware of it or not. An unprecedented number of commercial satellite launches [155] has recently led to the emergence of space systems as an unofficial category of critical infrastructure. These commercial systems provide high-speed, low-latency communications [104], position, navigation, and timing (PNT) [70], weather forecasting [108], military defense [153], and geospatial imagery [90], among other applications. Satellites range in size from centimeters to sizes capable of holding humans (for example, the International Space Station). Many satellites on the smaller end are run by universities through academic research labs or for-fun clubs run by undergraduates, while governments or commercial entities typically operate larger satellites.

All satellites big and small, commercial and noncommercial, make up this satellite ecosystem. By satellite ecosystem, I mean the set of satellites in all orbits in space, along with ground components used to send commands up and receive downlinked data. This ecosystem is intertwined into the lives of every person on the ground. Satellites underlie much of our critical infrastructure. Through GPS, the world navigates the seas and ground and also uses precise timing data of GPS for financial transactions. With or without our knowledge, sensing satellites collect shockingly high resolution images on the ground and even into backyards.

As the importance and prevalence of satellites continues to increase, so do the implications on the security, privacy, and safety of humans on Earth and in space. These risks are not limited to a particular type of satellite nor a particular adversary. Rather, they span the entire satellite ecosystem. For example, state adversaries may seek to damage

satellites launched by a competing country for financial gain, or to destroy a portion of satellite-facilitated critical infrastructure as part of cyber warfare. This threat to the satellite ecosystem is not theoretical; Russia attacked ground segments in Viasat's satellite communications system, leading to thousands of Ukrainian modems to go offline during Russia's invasion of Ukraine [106]. Other attacks on the communications link and ground stations have occurred, and there has been suspicion that actual satellite space components have also been targeted [126]. In addition, privacy threats exist given the impressive data collection capabilities of many satellites. Even without a security compromise, individuals could purchase increasingly affordable commercial satellite images for interpersonal surveillance, an emerging threat identified in this dissertation.

All of the threats listed above are important to consider, but why can't we just apply traditional security techniques used on the ground into the space domain? There are differences for both security and privacy challenges for satellites. We can take a drone as an example of a similar technology of an embedded system that flies and, like some satellites, can take images. First, consider the privacy concerns of drones. Like commercial imaging satellites, a drone can capture bird's eye view images any time each day. If someone flies a drone, they may likewise have access to raw, unedited images of the Earth. But unlike imaging satellites, most drones in the commercial market are detectable through at least sight, potentially alerting those who can see the drones of their presence and thus temporarily affecting how they behave while a drone is nearby. Next, consider the security concerns of drones and how they differ from satellites. Drones are a common platform used for testing sensor attacks and associated recovery techniques [34, 35, 98]. Compared to drones, satellites must handle a much longer mission, taking up to multiple years. Satellites must be able to handle very stealthy attacks that may only cause measurable impacts over periods longer than a single drone mission. A satellite's location must be precise to stay in the correct orbit while pointing its communications module and sensors in the correct direction, whereas a drone might be able to land many meters away from its target landing destination while still executing a safe mission.

Securing satellites poses difficult challenges. One issue is their possible vulnerability to an increased rate of hardware faults compared to terrestrial hardware due to radiation in orbit. In addition, the complexity of multiple control systems and sensors operating on a machine with limited power and computation to put towards attack detection makes building any solution to an onboard attack difficult. Satellites also face temporal and physical contact constraints. In low Earth orbit, satellites take around 90 minutes to orbit Earth. Single ground stations can typically only contact a satellite for 7-10 of those minutes [39]. And if the satellite were to enter an unrecoverable state, there is no feasible way to physically access the satellite to reboot it. Another challenge is latency constraints. Satellites may need to respond in real-time for tasks like collision-avoidance or ADCS [75]. Adding encryption and other security measures could increase the amount and size of packets needed when communicating. Yet another challenge is that many satellites are largely bespoke systems, with custom protocols, operating systems, and hardware components. This makes it difficult to use existing security tools, such as fuzzing [162]. Building defenses to any sort of threat onboard a satellite must also not violate timing constraints of a real-time system.

Past and concurrent academic work has enumerated and taxonomized threats to satellite security and the corresponding harms [32,43,45,73,83,84,103,126]. While many papers have focused on threat enumeration, more recently some work has taken a more experimental approach [57, 74, 75, 91, 107, 163] through building systems and analyzing existing satellite code. Yet there still remain gaps in the empirical analysis, threat enumeration, and towards finding solutions. Work in this dissertation addresses gaps in all three of these areas, described in more detail below.

Contributions

Below, I describe how this dissertation addresses satellite security and privacy research gaps. Then, I go over motivating threats and an overview for each chapter.

Research Gaps Addressed

Gap 1: Empirically Grounded Methodology for Satellite Ecosystem Protection In order to build better protections for satellites, we first need to deeply understand the barriers and challenges. Most of my work takes an empirical approach to understanding these challenges. Chapter 2 surveys people on how they weigh the benefits and possible privacy harms of commercial satellite imagery. These surveys, along with discussions with collaborators in geography, a panel of security and privacy experts, and ordering real satellite images, informed my proposals for future approaches to make satellite imagery useful while mitigating the harms to individual privacy. Chapter 3 also takes an empirical approach through a two-pronged methodology. Using a combination of in-depth interviews of satellite developers and analysis of their open source satellite code, Chapter 3 identifies practical barriers that small and amateur satellite developers face to building satellites securely. Chapter 4 likewise uses a hands-on, empirical approach, this time through a realistic satellite mission simulation. Through the simulation, this chapter characterizes the impact of a malicious satellite peripheral microcontroller on a flight computer, then shows that traditional defensive techniques do not work. Lastly, Chapter 4 builds a proof-of-concept technique atop the simulation to demonstrate a novel sensor diagnosis technique, aimed at identifying and then shutting off a malicious peripheral.

Gap 2: Underexplored Satellite Ecosystem Threats The empirical methodologies above aid in finding gaps in the threats listed in previous work. The results of expert panels and the survey itself in Chapter 2 identify possible individual privacy harms. Chapter 3 elicits emerging threat models during interviews. Chapter 4 identifies possible motivations and impact of a malicious peripheral, described further in the chapter overview below.

Gap 3: Towards Solutions to Satellite Ecosystem Threats Taken together, these empirically grounded results on barriers and challenges, along with threats to the satellite ecosystem that previously received little attention, provide a stronger foundation for

proposing and building better mitigations. Chapter 2 poses possible mitigations to privacy harms that can be used by commercial satellite imagery stakeholders. Chapters 3 and 4 discuss solutions to overcoming security barriers and to mitigating threats. Chapter 4 also builds a proof-of-concept solution that can be adapted to other satellites and cyber-physical systems.

Chapter Overviews

Next, I provide descriptions and contributions of each chapter.

Chapter 2: Privacy Threats and Concerns of Commercial Satellites Earth observation satellites have been around for decades, but they have expanded in availability and number in the past decade. The impressive data collection capabilities of these commercial satellites, along with their decreasing costs and increased availability to customers motivated Chapter 2. Commercial satellite imaging is used for diverse applications in a wide range of sectors, from agriculture to the military. As satellite images continue to become more widely available and detailed in resolution, the potential for individual and population-level monitoring increases and raises new privacy concerns compared to previous Earth observation technologies. These technologies may become even higher resolution in the near future. For example, a person that happens to become the subject of popular media may now face increased attention. With this attention, individuals or other entities might misuse satellite imagery to inspect images of their target's neighborhood, taken multiple times a day, to infer behavior and whereabouts.

To better understand privacy threats and concerns of commercial satellite imagery, this chapter reports on a survey of 99 participants from the United States. The survey found that most respondents were not aware that commercial imaging satellites existed, and once informed about the capabilities of commercial satellites, most are not comfortable with how good the current state-of-the-art satellite imaging capabilities are. Few respondents want satellite imagery cost-free and widely available, which conflicts with current trends in

geospatial data. In addition to aiding our understanding of the public's current perception and relationship with remote sensing technologies, this chapter uses the survey results to propose possible new satellite image legislation, regulation, and technological mitigations that can serve as discussion for stakeholders beyond this dissertation.

Chapter 3: Security of Small and Amateur Satellites This chapter pivots to a security lens of the satellite ecosystem. This chapter is motivated by the increasing number of satellite clubs at universities that are launching their own small satellites (smallsats). These clubs launch up handcrafted satellites with other academic projects, or their own for-fun payloads, attached to them. At first impression, it might seem unwarranted to focus on securing small satellites that have less capabilities and monetary value than more complex satellites. But there are multiple reasons as to why their security should be considered. Firstly, these satellite clubs are a training ground for those who enter the space industry, so security practices and perceptions of satellite clubs will to an extent inform security in industry. Second, these satellites can be used as a kinetic weapon to damage other objects in orbit. For example, what if a state actor compromised a United States university satellite with propulsion and then caused it to knock into the International Space Station, then blamed it on the country of the satellite club for not properly controlling their satellites? This and more threats are surfaced in Chapter 3.

This chapter explores the security practices and preferences of small satellite teams, particularly university satellite teams, to understand what barriers exist to building satellites securely. Through a combination of interviews and code audits, this chapter finds that security practices vary widely across teams, but all teams studied had vulnerabilities available to an unprivileged, ground-based attacker. Interview participants foresee many risks of unsecured small satellites and indicate security shortcomings in industry and government. Lastly, this chapter identifies a set of considerations for how to build future small satellites securely, in amateur organizations and beyond.

Chapter 4: Security Impacts and Defenses of Compromised Satellite Peripherals
As society increases its reliance on satellites as part of critical infrastructure, the risk of attack only increases, particularly from high-resource adversaries. At the same time, satellite fleets and building satellites at scale are becoming more popular. More fleets and scaled production means more shared hardware and software across hundreds of satellites or more. This increases the attack impact—and thus, attack likelihood—for attacks aimed at satellite hardware. Suppose there is a high-resource adversary that wants to target many satellites all at once. The adversary could sell a malicious peripheral to a startup at an affordable price purportedly in exchange for gaining flight heritage in space, a common practice in the aerospace industry. Once the adversary succeeds in having these malicious peripherals on board, the malicious peripheral code could be triggered under some condition while in space to achieve negative physical effects.

Chapter 4 demonstrates that common satellite architectures today are poorly equipped to handle malicious peripherals. Using NASA's popular open source core Flight System software (cFS) with current software and hardware reliability techniques, a system designer will inevitably confront a dilemma: Either the system deploys countermeasures against malicious components and suffers degraded nominal performance, or the system cannot survive malicious components.

Next, the chapter proposes a new sensor diagnosis technique, called *ControlPhoenix*, aimed at being able to identify and shut off a malicious peripheral on a satellite system. We show how *ControlPhoenix* can be applied to other cyber-physical systems as well, discussed more in-depth in the chapter.

Together, Chapters 2-4 fill in threat and defense gaps across the satellite ecosystem, making it safer for all. Chapter 5 summarizes these contributions and concludes.

Chapter 2

Privacy Threats and Concerns of Commercial Satellites

This dissertation first explores the implications of increased availability and decreased cost of commercial satellite imagery. Individuals are now allowed to buy images with little oversight, as my collaborators and I empirically confirmed through ordering multiple images. The ease of access, combined with high spatiotemporal resolution, leads to new questions about the potential for individual surveillance by non-government entities. Chapter 2 explores these emerging threats through a survey of 99 participants in the United States, followed by a discussion of possible regulatory, legislative, and technical approaches towards mitigating these threats. This chapter initially appeared as a paper titled "Over Fences and Into Yards: Privacy Threats and Concerns of Commercial Satellites" at Privacy Enhancing Technologies Symposium in 2024 [112]. Results are current to the time of the study in 2023.

2.1 Motivation

Space exploration in the 20th century opened the ability to observe and measure Earth from a completely new perspective. While offering numerous benefits, this vantage above the planet also creates new potential avenues of human surveillance, including satellite imagery, which can surveil the Earth's entire surface. As satellite imagery resolution continues to improve, its providers have also expanded access to these images beyond just governments. Since the 1990s, many governments, including the United States, have permitted commercial satellite companies to sell imagery to both governmental and non-governmental customers.

Commercial satellite companies collect images from satellites and sell these images to customers, and can even sell to individual people. These customers use the images for a variety of applications, including agriculture, scientific research, identifying illegal activity

(e.g., cannabis farming and illegal, unreported, and unregulated (IUU) fishing), defense, and even finance [131]. International and U.S. space law embody a web of complex legislation [114] that is continuously evolving [157], but U.S.-based restrictions on spatial resolution of imaging devices have relaxed in recent years.

Compared to platforms like Google Earth and Google Maps, which purchase and sometimes post-process commercial satellite and aerial imagery, direct-to-consumer commercial satellite imagery introduces unique privacy and security challenges. In addition to being able to purchase unaltered images on-demand directly from a commercial satellite company, another key difference is the temporal resolution of commercial satellite companies, or the frequency at which they image: whereas Google Earth provides undated high-resolution images that are not frequently updated, commercial firms collect imagery of any given location of Earth up to multiple times a day, sometimes dependent on customer requests [127, 128]. This high temporal resolution can allow individuals to be tracked across time at a higher granularity. Abstractly, drones might represent a similar technology to satellite imagery in that they can capture bird's eye view images any time each day. If someone flies a drone, they may likewise have access to raw, unedited images of the Earth. But unlike satellites, most drones in the commercial market are detectable through at least sight, potentially alerting those who can see the drones of their presence and thus temporarily affecting how they behave while a drone is nearby (discussed further in Section 2.2).

This chapter is motivated by the observation that (1) commercial satellite image resolution continues to improve in spatial, temporal, and spectral resolution (the minimum portion of the electromagnetic spectrum that it can resolve), (2) these images are simultaneously becoming cheaper and more widely available, which has important implications to individual privacy, and (3) satellite images might *not* be post-processed from a privacy-protecting and anti-surveillance perspective. Our research is the result of a cross-disciplinary collaboration spanning computer security and privacy (two authors), social psychology (one author), and political geography with a focus on remote sensing (one author). We aim to understand what current and future individual privacy threats

commercial satellites pose, so that we can better consider possible mitigations to these privacy risks. It is important to understand the risks before commercial satellite capabilities improve further.

To guide our study and analyses, we formulated two research questions. Our first research question is:

- RQ1: What factors influence privacy versus utility considerations of satellite imagery?

As noted earlier, there are numerous commercial satellite imagery use cases that offer significant positive benefits. However, there are also privacy risks. RQ1 is designed to help understand how people consider and weigh privacy versus utility, and what those tensions between privacy and utility are. Foreshadowing our results, and as we hypothesized at the outset of this project, we find privacy and utility to be in tension both within individual respondents' reasoning and between respondents. Given the significant investment in commercial satellite imagery, we believe that it is inevitable that commercial satellite imagery will continue to decrease in cost and increase in spatial and temporal resolution. Thus, absent significant forethought, the risks to people of privacy invasion and surveillance may heighten. Ultimately, our goal is not simply to acknowledge that tensions exist between privacy and utility, and that the risks of privacy and surveillance harms may increase in the future, but to provide guidance on how to navigate those tensions. With this in mind, and informed by our findings with respect to RQ1, we formulate our second research question below:

- RQ2: What possible regulatory, legislative, and technological approaches are there to protect individual privacy against commercial satellite imagery, and how do those approaches address the factors and tensions identified in RQ1?

To answer our research questions, we conducted a survey of 99 U.S. residents. We focus on respondents in the U.S. so that we can understand possible U.S.-specific legislation recommendations, though many of these could extend internationally, particularly as many technological standards set in the U.S. end up being adopted in other national

and international contexts. The U.S. has also already demonstrated initiative in setting pioneering regulations to govern satellites, as with its commitment to not perform direct-ascent-anti-satellite missile testing [67] promulgated in 2021.

Our survey responses directly inform our analysis of RQ1. Elements of our survey also contribute directly to our answer to RQ2, though we reach beyond just our survey's responses as we answer RQ2. Namely, our answer to RQ2 is a result of the synthesis of our survey's answers, additional reflection, and the perspectives brought by our diverse backgrounds (political geography, psychology, and computer security and privacy). We pose key considerations for legislators and regulators, as well as possible technological protections to individual privacy (Section 2.7). By raising awareness among researchers and practitioners, we hope that future research and communication with the public will help inform future decisions in how to manage commercial remote sensing technologies as they proliferate worldwide.

2.2 Background and Related Work

Spatial and Temporal Resolution. Spatial resolution is the smallest dimension on the Earth's surface represented by one pixel in an image, e.g., a 3m spatial resolution means that one pixel in the captured satellite image represents a 3 x 3m square of the Earth's surface. Temporal resolution is the frequency at which images are taken. A 1 day temporal resolution means satellite images are taken of a particular location on Earth up to once per day.

Satellite Capabilities. Commercial Earth observation satellites have been on the market since the late 1980s [147], and are now offered by a variety of companies. Commercial satellites' spatial resolution has drastically improved from tens of meters to now centimeters, with some commercial satellite companies boasting resolutions of 15cm [51], shown in Figure 2.1. Public documents suggest that U.S. government military satellites may offer 10cm resolution or better [121]. Thus, there is potential for commercial satellite companies to offer higher resolution in the future. Some satellite imagery provides both high spatial and temporal

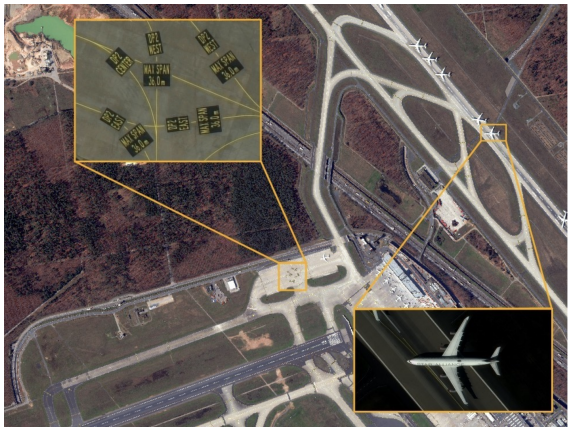


Figure 2.1: Example of 15cm resolution satellite image from Maxar [51]

resolution; satellite "videos" stitch discrete satellite images into videos that last between 30 seconds and two minutes [132]. Although not available yet, startup EarthNow aims to create a real-time video stream of the whole Earth [6, 14]. For satellites taking more than a short video clip, temporal frequency capabilities are lower, but are still high at up to 15 times per day [109].

In addition to visible wavelengths, satellite imagery captures other spectral bands. For example, healthy vegetation reflects near infrared wavelengths, which can be measured to help determine crop health and deforestation [130]. There are also different forms of radar, including Synthetic Aperture Radar (SAR). SAR imaging allows for observation at night and cloud cover penetration [50], making it highly useful in military and surveillance contexts.

While many satellite images are taken at an angle perpendicular to the ground, some are also capable of imaging many degrees off this angle, known as "off-nadir". We include a real example in Figure 2.2 to demonstrate the capabilities of this type of spaceborne imaging. As a first work to study privacy and utility tensions with commercial satellite imagery, we focus on today's most common commercial satellite image form: perpendicular to the Earth.

Related Technologies. Prior work has raised privacy concerns in related technologies that can measure images from a bird's eye view, including drones, Google Earth and Google

Figure 2.2: Example of a 60° off-nadir satellite image from Planet Labs [82]

Street View. Google Earth provides the public with free, undated, high-resolution overhead images of most locations on Earth. Google Earth Pro, with its timeslider function, offers more historical satellite imagery, yet it still is typically only available at time intervals of one year apart or longer and thus is less of an individual privacy threat than purchasing commercial satellite images at a high temporal resolution.

In the U.S, there are concerns that Google Earth and other similar services violate reasonable expectations of privacy [94]. In France, the French government used Google Earth data and artificial intelligence to detect undeclared pools that were circumventing French tax laws in residents' backyards in 2022 [21]. To foreshadow our findings in Section 2.6.4, many people were uncomfortable with imaging backyards, even to capture illegal activity. Although these practices may become more commonplace, some may see them as an invasion of privacy, which will be important for regulators, legislators, and commercial satellite providers to consider.

Multiple works explore how users view drones as a privacy concern [24, 159]. Concerns such as drones recording without consent [24] and stalking [159] also apply to potential surveillance through commercial satellites. Drones are similar to satellites in that they can take high-resolution bird's eye images. From the perspective of an adversary, using drones and satellites to violate individual privacy have different tradeoffs. Drones require a one-time purchase and will cost less money upon taking multiple images compared to current

satellite imagery prices. Most drones would be detectable while in the air, at least visually, and thus the adversary would risk exposing their presence. With satellites, most people are not aware of when one is taking images over their area. Satellite images are currently lower resolution than drones but with time, the difference between their spatial resolutions may decrease. Further, whereas a purchaser of a commercial satellite image does not need to be physically near the surveilled location, the pilot of a drone (or at least the drone itself) must be physically in the same vicinity.

In Google Street View, there have been privacy concerns and efforts to blur out humans, license plates, and other personal information [49, 53, 119]. Google allows people to request to blur their face or home in Google Street view [60]. It is unclear what satellite companies blur and how it differs by company. Even still, the update frequency of Street View images is lower than the temporal resolution capabilities of commercial satellites.

Commercial Satellites Security and Privacy. Privacy concerns surrounding commercial satellite imagery have existed for decades and remain a topic of concern [14, 124, 134, 147]. Atluri et al. construct an authorization model for those using satellite data, which provides different spatial resolutions for different authorization levels [10]. Black discusses privacy concerns of commercial satellite imagery [18], though the focus is more on protecting privacy of U.S. military efforts and protecting national security than individual privacy. Concrete examples of satellites in use have affected other countries' privacy and have benefited the U.S.: an off-nadir photo was used to discover a North Korean submarine docked in North Korea that was hidden from perpendicular angle photos by an awning [144]. To our knowledge, no previous research has deeply investigated individual privacy concerns for commercial satellite imagery; our work fills this gap.

Democratization of Access to Satellite Imagery. There are movements in industry and the geography field overall to democratize access to imagery, i.e., to make satellite images more widely available and lower cost (and even no-cost). Skyfi, a startup whose app launched in early 2023 [172], states that its mission is to "democratize satellite film and imaging. What

was once a high-tech process is now simple, affordable and accessible to everyone." [145].

Many geographers support democratization efforts, such as aiding in armed conflicts [15] and scientific research [166]. The Consortium for Execution of Rendezvous and Servicing Operations (CONFERS) [30] recommends that "the U.S. government should declare the space domain as a public space and the ability to conduct [imaging from space] as the equivalent of taking photos of public activities on a public street" [29].

Satellite Regulation/Legislation. Laws surrounding remote sensing are complex, and have even been described as "byzantine" [114]. They also have changed dramatically over the past several decades. While the U.S. government previously limited spatial resolution of U.S.-based commercial satellite companies to only have as good of spatial resolution as the best resolution of foreign competitors [141], the Licensing of Private Remote Sensing Space Systems rule of 2020 removed this restriction [122]. Then, in 2023, in a move to ensure the U.S. commercial satellite sector's continued international dominance, the National Oceanic and Atmospheric Administration's Commercial Remote Sensing Regulatory Affairs office ended temporary restrictions which had limited rapid revisit over certain areas. The regulatory change also enables imaging and distribution for 99% of the Earth's surface and removes previous limitations on X-Band Synthetic Aperture Radar capabilities [76]. There is also an ever-growing number of foreign commercial satellite companies that are not governed by U.S. restrictions, such as China's GaoFen satellites and France's SPOT Images [151] (with a previous spot satellite sold in 2014 to Azerbaijan [37]), to name just a few.

2.3 Pre-Survey Methodology

Before creating our survey on people's threat perceptions and concerns, we took the following steps:

1. Conducted a brainstorming exercise among security and privacy experts and non-experts to generate a set of privacy concerns of satellite imagery (Section 2.3.1);

2. Purchased real satellite images to better understand the current state of commercial satellite imagery processes (Section 2.3.2).

2.3.1 Brainstorming Exercise

We conducted a collaborative brainstorming exercise among 25 colleagues to explore the space of potential privacy threats and concerns of commercial satellite imagery. Our goal in understanding these threats was to inform the questions we constructed in our survey. We developed two separate codebooks based on these brainstorming sessions:

1. A private activity image metric codebook: aspects of images that if observed, could violate individual privacy (discussed in Section 2.3.1);
2. A privacy threats and concerns codebook: a summary of themes that emerged of types of activities that might violate privacy if imaged (discussed in Section 2.3.1).

Brainstorming Exercise Methodology

We discussed privacy concerns of commercial satellite imagery across multiple sessions where half of the volunteer colleagues were security and privacy experts; the other half were outside of the security and privacy field, including a geographer. This methodology was inspired by Hiniker et al [64]; we complement this expert panel with colleagues outside of security and privacy to offset possible biases within the security and privacy field. Our sessions took the form of individual brainstorming and writing of ideas, a group discussion to surface potentially even more ideas, or both in that order during the same brainstorming session. For all of our brainstorming sessions, we used the following two prompts: "What activities might someone do near their home that they do not want captured with satellite images? Please list as many as possible, including activities that you don't do but someone else might", and the second prompt was the same except it asked about activities *away from* someone's home. Lastly, one author went through a structured brainstorming threat modeling exercise based on brainstorming cards from Denning et. al [38], and then discussed these results with

a second author, who is an expert in security and privacy.

While the pre-study activities were not the focus of our research, we sought to maximize the potential of pre-study activities to positively influence the content and format of our actual study (Section 2.4). Thus, we adopted qualitative study best practices as we analyzed our pre-study data. Through notes and written responses collected from these brainstorming sessions, one author developed two initial codebooks and performed thematic analysis [20] of all ideas that were surfaced in the brainstorming sessions. The main coder updated and discussed with a second author throughout the process of coding, and also met once to discuss all notes collected during the process, as well as discussing coding and theme construction. We used this process for both codebooks.

Brainstorming Exercise Analysis

Both codebooks that we produce through our analysis are meant to be a guide for our study and not a generalizable taxonomy.

Privacy Threats and Concerns Codebook. We constructed an activities type codebook to generate content for a subset of questions in the user study. For this codebook, we coded for types of private activities that could potentially be captured by satellite imagery. From our brainstorming notes, we categorized everything into the following three activity types:

1. "Innocuous": activities that we believe many United States residents deem culturally acceptable and legal. Brainstorming session examples include going to a bank and shopping.
2. "Doxxing/Non-universal acceptance": activities which are legal in the United States, but might not be widely accepted in United States society. These include visiting sensitive locations such as a mental health clinic or going somewhere instead of work when claiming to be sick.
3. "Illegal": any activity that is illegal in at least some part of the United States, such as speeding or growing marijuana.

Private Activity Image Metric Codebook. We used the private activity image metric codebook to map private activities to aspects of images that would need to be observable in order to see that private activity. The metrics were based on the privacy concerns people brought up in the brainstorming session. The full codebook diagram is included in Appendix A.3 for reference. An example metric is being able to observe "people body position", from which someone can infer whether someone is doing a particular activity, such as running. Another example metric is "which car" which could be used to distinguish between cars and track someone's car location across time. We use this codebook to help construct some of our survey questions, such as asking about choosing between two scenarios that require being able to observe the same type of image metric, or metrics that require around the same spatial resolution to observe.

2.3.2 Ordering Commercial Satellite Images

To understand what the commercial satellite image purchasing process is like for civilians, we shopped for images from three U.S.-based vendors, which we anonymize as A, B and C, and one Chinese vendor, D. Vendor C was the least expensive and the easiest to purchase, so we only purchased from them. Purchasing on-demand 75cm spatial resolution images over our institution's campus from Vendor C cost around $175 each time. Vendor C did not require any human interaction. We ordered an image twice, and each time we received our photos within a couple days of requesting them. This simple process contrasts with two older companies, A and B, that we inquired. Company A offered a sales option of $15,000, and company B required a minimum purchase of $5000. Company A required filling out a form. They never responded to one author but did respond when another author requested photos separately. Company D offered on-demand 30cm spatial resolution imagery for a minimum of $5600, but the company was based in China and it was only possible to pay via WeChat; the complexity of getting permission to purchase from company D precluded us from proceeding. We expect that the trend of decreasing price and increasing ease of purchase of satellite images will continue, which also increases the possibility for people to

potentially misuse these purchases.

2.4 Survey Methodology

We developed a study to answer our underlying research questions RQ1 and RQ2. The entire survey is included in Appendix A.1 for reference. Through this survey, we sought to understand respondents' awareness, privacy perceptions, and privacy concerns surrounding commercial satellite imagery, as well as their perspectives on balancing privacy with utility. We conducted a pilot study of 15 participants. We read through and studied every response. Because we thought the participants were interpreting the questions how we expected, we did not adjust any of the survey structure following the pilot study.

2.4.1 Participants

We recruited participants from Amazon Mechanical Turk. We recruited them from only the United States because we wanted to understand how United States residents' responses and concerns compared to current United States commercial satellite regulation. To increase the likelihood of high-quality responses, we filtered for an Mturk approval rating of over 98% and an Mturk Masters Qualification. Through an a priori power analysis for a one-sample t-test using G*Power, we determined that 90 participants were needed to detect a small-moderate effect size (d = 0.30) at 80% power with a criterion of $\alpha = .05$. We oversampled to allow for possible exclusions based on inattention or falsely entering a code on MTurk. Our final results consisted of 99 complete survey responses. They were compensated $5 for an estimated survey time of 20 minutes. On average, participants took around 30 minutes to complete the survey. Other demographic details are included in Table 2.1.

2.4.2 Survey Structure

MTurk workers were told that we wanted to understand their thoughts about satellites. We purposely avoided discussion on privacy or harms to avoid priming them to think negatively

Table 2.1: Breakdown of participant demographics by gender, age, race/ethnicity, and political affiliation.

Gender		Age		Race/Ethnicity		US Political Affiliation	
Man	64.6%	25–34	16.2%	White	82.9%	Democrat	62.6%
Woman	35.4%	35–44	51.5%	Black or African American	8.1%	Independent	19.2%
		45–54	20.2%	Asian	5.1%	Republican	17.2%
		55–64	10.1%	Mixed	4.0%	Libertarian	1.0%
		65+	2.0%				

about satellites. Participants first read the consent form including a description of the study. We then provided definitions of satellite-view imagery and filtered for understanding of these definitions by asking them to label three images as a "satellite view" image or not a "satellite view" image. Secondly, we filtered for visual impairment since some questions involve evaluating images. After filtering, there were 99 respondents.

Awareness and Initial Perceptions. Participants read a brief description of how commercial satellites are used today. To assess overall awareness, we asked whether people have heard about commercial satellites based on that definition. Next, to begin to evaluate how people weigh privacy risks versus utility, participants responded to two open-ended questions evaluating (1) what they think are the possible benefits of satellite images, and (2) what they think are the possible harms of satellite images. These questions came first in the survey to promote creativity in responses and to avoid priming with specific scenarios of privacy violations. We asked several other questions assessing awareness of satellite image practices, including what they thought the current spatial resolution, temporal resolution, and cost of satellite imagery is. For spatial resolution, we showed drone images that we took at different

spatial resolutions, ranging from 15cm (the highest commercial satellite resolution today) to 10m resolution (see Ethics Section 2.5 for our drone imaging practices).

Weighing Benefits and Privacy Harms. To help evaluate how people weigh personal privacy compared to the benefits satellite imagery could bring, we asked participants to consider possible pairs of scenarios. Participants could choose to allow either both scenarios or neither scenario, but not one or the other. This "would you rather" question formulation was inspired in part by the methodology from Simko et al. [143]. We chose to allow only "both" or "neither" responses to the scenario pair questions to require participants to choose whether they value privacy or utility more for the given scenario pair. We constructed scenarios based on the image privacy metric codebook, as discussed in Section 2.3 and presented in more detail in Appendix A.3. For each question, we paired a scenario with clear benefits and a scenario with privacy implications that both require a similar spatial resolution in order to be detected. For example, the first question contained two scenarios where the car presence metric is observed: one where a satellite would detect the existence of cars in hospital parking lots to estimate disease levels over time (beneficial), and one where an employer takes an image over their employee's home to ensure their car is there after the employee called in sick (privacy implications). To understand the factors involved in why people chose to allow both or neither scenario, we also asked them to explain each answer in a free response box. Table 2.2 presents each scenario pair, along with the results, which we return to in Section 2.6.3.

Comfort. We asked additional questions to understand the factors that might influence people's consideration of privacy and utility. We gave participants examples of locations satellites might image, activities that satellites might image, and entities who might have access to satellite imagery. The majority of our examples were inspired by the concerns people brought up in the brainstorming session (Section 2.3.1). Participants were asked to sort each activity and location into whether they thought it was acceptable or unacceptable to be imaged by satellites. For entities, they were asked to sort each entity into whether that

entity should have access or not have access to satellite imagery. We aimed to evenly split locations and activities across the three activity types ("innocuous", "doxxing", "illegal") per our private activities codebook.

Next, we asked participants to choose a statement about satellite imagery access that they most agreed with out of four possible statements: (1) "Satellite images should be available to everyone for free", (2) "Access to satellite imagery should cost money and depend on what the user is requesting the imagery for", (3) "The U.S. government should be the only ones allowed to access satellite imagery", and (4) "No one should have access to satellite imagery". Statement (2) corresponds with the typical practices of commercial satellite imagery today, and Statement (1) corresponds with current trends in geospatial data (Section 2.2). We then showed a set of photos taken with a drone over a parking lot ranging from 1cm to 1m spatial resolution, and participants indicated the highest spatial resolution that they would be comfortable with being taken of them. We then asked what temporal resolution they were comfortable with assuming their previous choice of spatial resolution, ranging from "every second" to "never".

Demographic Section. We included an attention check right before the demographic section, which every participant passed. Full lists of options are included in the full survey instrument in Appendix A.1.

2.4.3 Analysis Techniques

For all free responses, we used grounded theory to first take shorthand notes of all conceptually new ideas people brought up, then sorted these into a set of more overarching themes. One author proposed a codebook based on these notes, and then two additional authors (along with the first) discussed themes until consensus was reached. The codebook can be found in Appendix A.2. Two authors coded each qualitative analysis for which we discussed results, and had an inter-rater reliability of 95.5% agreement. Cohen's kappa was $k = 0.885$, indicating strong agreement [28].

2.4.4 Limitations

Crowdsource responses do not necessarily represent society as a whole, or even a specific region, like the United States. Rather, crowdsource responses at most represent the population participating in the crowdsourcing platform. Additional considerations and tensions might arise if studying other demographic groups. We recommend the study of additional populations in the future, including residents outside of the United States and historically marginalized populations.

MTurk is the only crowdsourcing platform allowed for use by our institution at this time, and hence we used MTurk. While we were initially skeptical about the use of MTurk, given prior anecdotes as well as our own prior experience with MTurk, all authors were impressed by the quality, relevance, and thoughtfulness of all free-responses in our survey. To increase the likelihood of high quality responses, we filtered for above 98% approval ratings on MTurk. To further indicate that participants were engaged, every participant passed the attention check at the end of the survey, which we designed to not be obvious for those skimming. Since there was not an option on MTurk to choose a demographically representative sample, the MTurk demographics were not representative of the overall demographic proportions in the United States. Our respondents were skewed toward men, white, and Democrat political affiliation.

For the drone images we showed participants in the survey, we could not show past 5cm spatial resolution on some images due to drone image quality at the height needed to capture the necessary physical space. This limited what set of images we could show participants, while attesting to the impressiveness of the high spatial resolutions of commercial satellite imagery.

2.5 Ethical Considerations

Performing the survey was determined to be exempt by our institution's IRB. Our participants were kept anonymous, and the most sensitive questions we asked were the

demographics questions. Our IRB had access to a summary of the survey protocol. Taking drone images to include in our survey was determined to not be human subjects research by the IRB. We obtained all necessary permissions for using drones and followed all relevant Federal Aviation Administration (FAA) regulations. We only took drone images in confined spaces, so no unintentional bystanders' privacy was violated while imaging. We provided each car owner and human participant an individual waiver to sign.

2.6 Satellite Survey Results

We now turn to presenting our results. For each subsection of results, we include key considerations that we incorporate into our suggestions for future satellite imagery practices (Section 2.7). Analysis of demographic differences is included in Section 2.6.6.

We present descriptive data and statistical analyses in order to describe the factors influencing privacy versus utility considerations for the sample studied here. We do not suggest all populations would have the same responses. Instead, even if these considerations and tensions only apply to a subset of people, those people are important and must be considered.

2.6.1 Satellite Image Awareness

Before turning to our specific research questions, we first sought to establish a baseline understanding of the degree to which respondents are aware of satellite images and their practices. Only 32% of respondents knew that commercial imaging satellites existed. Given a set of options between once a second and once a year, 58% of respondents thought that a satellite company could image the same location every day at most (either daily, weekly, monthly, or yearly). As stated in Section 2.2, some commercial satellites have already exceeded this once per day capability; some can image locations up to 15 times a day [109] and others can create a real-time video for up to two minutes [132]. Given a set of images of resolution options, 91% of respondents correctly guessed that the best commercial satellite

spatial resolution is around 15cm. This could be because 15cm was the highest resolution option we presented to participants due to drone imaging limitations.

Guesses about how much one satellite image costs ranged from $US1-20,000,000. The median guess was $100, which is similar to the current cheapest prices; it cost us around $175 for a 75cm resolution image of over 9 square miles (Section 2.3.2), but the cheapest imaging that were less recent and lower resolution were under $50.

Consideration 1: Lack of awareness of temporal resolution capabilities of satellite imagery means some privacy expectations do not align with possible privacy violations.

2.6.2 Benefits and Harms

We next explore how respondents discuss the benefits and harms of satellite images to begin to answer RQ1. Recall from Section 2.4 that to avoid priming, we ask about benefits before harms and do not mention privacy.

We first examine the harms participants considered, and find that the majority of participants consider the privacy risks of this technology in their responses. In particular, 52% of respondents explicitly used the words "private" and "privacy" in their responses as potential harms. Overall, our qualitative analysis (described in Section 2.4.3) led to the following categories of *harms*:

- **Invasion of privacy**. This includes monitoring day-to-day activities or specifically on personal property.
- **Security**. Participant 17 (P17) believed that satellite images "being in the possession of third parties means that any information that can be captured by a commercial satellite is no longer safe, secure, or private."
- **Facilitating Crimes**. Respondents mention facilitating both terrorism and burglaries.
- **Military Operations**. One respondent mentioned that it could help the military plan out a bombing.

- **Surveillance**. This theme includes monitoring of activities by the US government, foreign governments, and civilians. Stalking was specifically mentioned as a potential risk.
- **Law enforcement Abuse**. P14 was concerned that law enforcement "could get images without having to get a warrant."
- **Natural Resource Misuse**. Specifically, companies using satellite imagery to find natural resources.
- **Unwanted Advertising**. Companies could target advertisements based on what is in someone's backyard, such as a pool-cleaning service for pool owners.
- **Incorrect Identification**. Misidentifying items based on satellite images.
- **Compliance**. An example is "homeowners being fined for the condition or reshaping of their roofs" (P53). As seen in Section 2.2, the French government has already carried out one such application, fining its citizens for unregistered pools.
- **AI Integrations**. This means using AI to facilitate the above harms. Only one respondent mentioned this.

Possible benefits of satellites mentioned tended to be more varied and specific than the harms posed. It is clear that respondents were able to consider the possible utility of these satellite images along with their harms. We split *benefits* into the following categories:

- **Mapping**. 21% of people mentioned mapping use cases as a potential benefit, such as for improved map accuracy, logistics, or "to see if an attraction is crowded so we know to go another day" (P66).
- **Real Estate and Infrastructure**. Providing additional information for purchasing real estate, facilitating city planning, evaluating housing density, and understanding traffic patterns are some examples people mentioned. 35% of respondents mentioned this type of benefit.
- **Government Accountability**. Some respondents mentioned how more access helped prevent government abuse, including this response: "deforestation ramped

up drastically in Brazil during Bolsonaro's reign. I think satellite images from commercial enterprises could keep people informed so maybe they could put pressure on governments" (P14).

- **Legal Compliance and Crime Detection**. P79 explained that some of their previous tasks on MTurk were to evaluate satellite imagery, including looking for illegal oil refineries.
- **Targeted Advertising**. Some respondents discussed how looking at people's backyards could help companies sell products, such as solar panels.
- **Emergency Response**. Respondents mentioned how satellite images could help locate missing entities such as ships, people, and cars. P79 explained that some of their previous tasks on MTurk were to find "illegal oil refineries, whales and even that missing Malaysian airliner". Others mentioned using images to check structures for damage or respond to natural disasters.
- **Studying Natural Phenomena**. Examples include climate change, biodiversity, and volcanoes.
- **More Information**. This includes information in general and not about a specific topic, as P45 states: "Any data is good data. The more I see, the more I know".
- **Surveillance**. Either government or civilian entities collecting information about government or non-government entities.
- **Agriculture**. Examples given include planning out land use and monitoring crops.

Some benefits and harms overlap, highlighting how privacy and utility can be in tension between respondents. For example, advertising, government surveillance and information gathering all fall under both potential benefits and harms.

Consideration 2: Privacy violation is a concern of potential satellite harms.

Consideration 3: There is disagreement between whether some applications are harmful or beneficial.

2.6.3 Privacy versus Societal Benefits of Satellites

We further investigate RQ1 through analysis of the would-you-rather questions. Quantitative analysis of participants' choices, along with qualitative analysis of their explanations, allowed us to more deeply explore how they weigh privacy versus utility, and whether there are tensions between the two. The results, along with the scenario questions, are summarized in Table 2.2. For each pair of scenario questions, we ask "Would you rather satellites have both of these capabilities, or neither of these capabilities?".

Scenarios Question 1

This first question addresses activities that are measurable from satellites images if car presence is observable in the images: "Consider scenarios A and B. A: Satellite images are used to track the number of cars at a hospital, which helps with predicting levels of disease in the area. B: Someone calls in sick to work. To check whether they are at home, their employer takes a satellite image over their home and observes whether their car is there." Only 21% chose to allow both. An example of why someone chose both is for "the greater good" (P60). Others cited that there is no reasonable expectation of privacy: "Public is public. Get a garage if you don't like it" (P79). Of the 79% who chose that neither should be allowed, many deemed monitoring employees unethical or the privacy violation to be too high. For example, one response was that "The benefits of A do not outweigh the invasion of privacy afforded by B" (P17).

Scenarios Question 2

Scenarios in this question require observing the presence of humans or objects, with scenario D posing monetary benefits: "C: Someone wants to know whether their former romantic

Table 2.2: Scenarios with individual or societal benefits versus scenarios that violate privacy. Percent of respondents who chose to allow both, organized by demographic. Full scenario questions are included in Section 2.6.3.

Scenario Pairs	Overall	Women	Men	Democrat	Republican
Scenario A: Detecting cars in hospital to track disease. Scenario B: Employer checks employee's car who calls in sick.	21%	14%	25%	19%	35%
Scenario C: Former romantic partner looking into backyard for visitors. Scenario D: Confirm items lost during fire for insurance.	26%	26%	27%	16%	65%
Scenario E: Detecting celebrity's activities. Scenario F: Locating a missing hiker.	57%	60%	56%	45%	94%

partner is having any visitors to their house, and uses a satellite image to detect how many people are gathered in their former romantic partner's backyard. D: Someone's backyard catches fire, and their insurance provider uses a satellite image to confirm the items that were lost due to the fire, which ends up giving the fire victim more money than expected". Only 26% chose to allow both. Someone didn't believe that the privacy infringement could hurt the victim: "The spying in this case will not lead to a privacy infringement that can hurt someone and the fire damage issue is relevent" (P3). Another respondent argued that "Gathering information about an ex and settling insurance claims are both important" (P84). The 74% who chose to deny both cited the threat of stalking and/or spying. Others cited their personal stance on privacy: "I would rather give up the ability of an insurance company to better pay out losses than give out private information" (P25).

Scenarios Question 3

Both scenarios in the third question require observing and identifying individuals: "E: Satellite images are used to detect what a well-known celebrity is doing in their backyard. F: Satellite images are used to locate a hiker that went missing in the mountains." 58% of respondents chose to allow both scenarios, with reasoning such as the value of saving a human life, as well as the current privacy expectations of celebrities: "A celebrity does not expect privacy" (P84). Some of those who disagreed mentioned the high value of privacy, such as: "Satellite images to find a hiker would be good, but not if it comes with violations of other people's privacy" (P14).

Scenario Question 4

We included a fourth scenario question that wasn't as closely tied to individual privacy asking about using satellite imagery for conservation versus people using it to find species for illegal hunting. There was overall 53% agreement to allow both. This question was used as reference to compare with the scenario pairs that tied more to individual privacy. This fourth question was the most evenly split in responses across all scenarios.

Scenario Commonalities

Across all scenarios, there were several commonalities in how people weighed privacy concerns versus potential benefits. For example, respondents justified allowing both or neither scenarios by saying that there are better and cheaper ways to achieve the same goal. Respondents cited privacy in choosing either option for each scenario, saying that the social benefits of a scenario are more important than privacy or vice versa.

Consideration 4: There exist temporal and spatial resolutions and application domains such that there is no consensus on whether that resolution of satellite imagery should be allowed.

Consideration 5: Whether people prefer satellites to be allowable at a given resolution depends heavily on the use case in question. If not possible to conditionally choose which satellite use cases are allowable, potential privacy threats may outweigh societal benefits of satellites at certain resolutions.

2.6.4 Comfort with Satellite Imagery

We gauged the commercial satellite capabilities and use cases that respondents were comfortable with to further understand what factors influence privacy versus utility tradeoffs (RQ1).

Resolution Capabilities

We showed respondents drone images we took of volunteers that were downsampled at different spatial resolutions, ranging from 1cm to 1m. Most respondents said they were comfortable with images being taken of them between 1cm and 15cm, which is equivalent or higher than current commercial satellite capabilities [51]. However, 84% of respondents were not comfortable with these spatial resolutions being taken more than once a day. And

36% of respondents were not comfortable with the images being taken more than once a year. In other words, people are comfortable with current spatial resolution capabilities but *not* when combined with current temporal resolution capabilities. See Appendix A.4 for a summary of resolution comfort results.

In general the higher the temporal resolution comfort, the higher the spatial resolution comfort as well. We expected that those who chose higher spatial resolution would choose a lower temporal resolution, but these results show that this is not the case. We hypothesize that this is simply due to people's general privacy preferences or expectations: someone who has lower privacy expectations or less of a preference for privacy is comfortable with having both high spatial and temporal resolution images taken of them.

Consideration 6: Many people are not comfortable combining very high spatial resolution with very high temporal resolution. Temporal resolution and spatial resolution should be considered in unison.

Consideration 7: Those who are comfortable with higher temporal resolution also tend to be more comfortable with higher spatial resolution.

Use Cases

Results from the bucket-sorting activity further illustrate how different factors, such as the use case, affect evaluations of privacy. Tables 2.3 to 2.5 summarize these results, showing what percent of respondents deemed each bucket acceptable to image or be imaged by.

The bucket deemed least acceptable on average was imaging people while sunbathing (4%), and the most acceptable was imaging a rainforest (96%). Even in the case of illegal activities, not everyone thought it was acceptable for satellites to image them; only 19% overall found it acceptable to image people doing illegal drugs in their backyard. 64% thought it was acceptable to image people doing illegal drugs in public, which attests to the privacy expectation people have in their backyards. Overall, only 11% thought it was acceptable

Table 2.3: Percent of respondents who think it is acceptable for entities to have access to satellite images. Overall percentage along with percentages by demographic (political affiliation and gender). W is women, M is men, D is Democrat, R is Republican.

Entity	Overall	W	M	D	R
Agriculture Industry	82%	74%	86%	76%	100%
US Government	79%	71%	83%	77%	82%
Police	71%	69%	72%	61%	94%
Non-profits	61%	51%	66%	56%	71%
World governments	47%	23%	61%	53%	35%
Finance industry	37%	29%	42%	32%	53%
Employers	19%	9%	25%	16%	41%
Romantic partners	16%	6%	22%	8%	47%
Children	8%	6%	9%	8%	6%
Convicted criminals	7%	6%	8%	6%	6%

Table 2.4: Percent of respondents who think it is acceptable for locations to be imaged by a satellite.

Location	Overall	W	M	D	R
Rainforest	96%	91%	98%	97%	94%
Illegal business	84%	77%	88%	81%	94%
Grocery store	59%	37%	70%	55%	76%
Native lands	54%	46%	58%	50%	71%
The White House	53%	57%	50%	48%	53%
Religious center	33%	23%	39%	32%	47%
Adult-only store	31%	17%	39%	27%	59%
Mental health clinic	28%	17%	34%	24%	47%
Playground	15%	20%	13%	6%	29%
Backyards	11%	6%	14%	8%	24%

Table 2.5: Percent of respondents who think it is acceptable for activities to be imaged by a satellite.

Activity	Overall	W	M	D	R
Car accidents	95%	94%	95%	92%	100%
Illegal fishing	81%	74%	84%	85%	76%
Building w/o permit	78%	74%	84%	85%	76%
Driving	67%	60%	70%	66%	59%
Illegal drugs (public)	64%	60%	66%	56%	76%
Visiting jewelry store	26%	26%	27%	21%	47%
Illegal drugs (backyard)	19%	20%	19%	15%	35%
Extramarital affairs	12%	9%	14%	6%	35%
Meeting friends & family	9%	6%	11%	6%	18%
Sunbathing	4%	3%	5%	3%	12%

to image backyards. Whereas legal activities that could be seen as doxxing material was on average acceptable by 33% of respondents, imaging illegal activities was on average acceptable by 65%. In fact, satellites are already used to detect illegal activity, such as IIU fishing [131], and as one respondent said, illegal oil refining. In the "innocuous" category of activities, imaging of satellites was still not deemed acceptable by all respondents. On average, these activities were deemed acceptable by 46% of respondents.

Consideration 8: Acceptability of imaging varies by use case, but there is no universal agreement on any category of activity, whether or not it's illegal.

2.6.5 Should We Democratize Satellite Image Access?

We begin to explore RQ2 by investigating respondents' preferences for regulatory or legislative approaches to protect individual privacy against commercial satellite imagery. Results are summarized in Table 2.6. 68% respondents most agreed with Statement 2, that access to satellite imagery should cost money and depend on the use case, which is the common-practice today. Only 12% agreed with making imagery widely available; this low agreement with Statement 1 goes against movements from the geography field to increase image access [15, 29, 30, 166]. The remaining 20% didn't want commercial satellites to be allowed at all, either choosing that only the U.S. government should have access or no one should have access. Some factors respondents mentioned in making their decisions were:

- Whether there could be conditional access, such as not being able to image private property.
- The societal benefits outweighing the harms (and vice versa).
- Privacy expectations.
- The value of information.
- Which entity a respondent does not trust.

Table 2.6: Agreement of different satellite image access statements, per Section 2.6.5.

Scenario Pairs	Overall	Women	Men	Democrat	Republican
Statement 1: Satellite images should be available to everyone for free.	12%	3%	17%	11%	12%
Statement 2: Access to satellite imagery should cost money and depend on what the user is requesting the imagery for.	68%	66%	69%	65%	88%
Statement 3: The U.S. government should be the only ones allowed to access satellite imagery.	13%	20%	9%	15%	0%
Statement 4: No one should have access to satellite imagery.	7%	11%	5%	10%	0%

- Preventing abuse.
- Fairness. For example, "let the market value it fairly" (P62). Others thought it would be most fair to give everyone access so that it was not only accessible to the rich and powerful.

Consideration 9: Some people believe that we should regulate who has access to satellites by charging money for images or granting access based on use case.

Consideration 10: Some people believe there should be additional specifications and rules in place to prevent misuse of satellite imagery.

Consideration 11: Some people believe that paywalls on imagery may prevent abuse by civilians but promote abuse by the government and other powerful and wealthy entities.

2.6.6 Demographic Differences in Responses

On an exploratory basis, we also investigated how respondents' demographics might impact their perceptions of satellite imagery. Before suggesting legislative, regulatory, and technical solutions, we wanted to understand whether differences in privacy preferences vary significantly by people's lived experiences and personal ideologies. If one groups' lived experience leads them to experience a privacy harm much more than another, then this difference should be considered.

We chose to investigate demographic differences by gender and political affiliation because we had theoretical reason to believe there might be relevant experiences or ideologies that affect perceptions of privacy amongst these groups [13, 152]. Within political affiliation, we compared between Democrats and Republicans. We excluded Independents from analyses because this group may lack a common ideology, and we had less theoretical motivation to investigate this group. Within gender, we compared between men and women only since

there were no responses for other genders.

Commercial Satellite Imagery Awareness by Gender and Political Affiliation. 39% of men claimed to be aware of commercial satellites before taking the survey, versus only 20% of women. 29% of Democrats and 35% of Republicans were aware. We ran 2 independent-samples t-tests total to understand differences in acceptance labels across demographics.

Use Case Acceptability by Gender and Political Affiliation. Tables 2.3 to 2.5 show use case responses organized by demographics. There are notable differences in responses by gender and by political affiliation. Overall, men found 48% of all buckets acceptable, compared to women who found 39% of buckets acceptable. We performed a t-test and found that the average acceptance level between genders across all buckets was statistically significantly different ($p < 0.05$). Some use cases produced especially pronounced gender differences in acceptability. For instance, 70% of men found imaging grocery store parking lots acceptable versus only 37% of women. While we do not know the cause, it could be because that among heterosexual couples, women spend more time grocery shopping and thus would me more affected by imaging grocery stores [140]. The most pronounced difference in entities deemed appropriate was that 61% of men thought it was acceptable for other world governments to access this imagery, as opposed to just 23% of women.

Republicans on average found 56% of buckets acceptable, while Democrats found 42% of buckets acceptable. We performed a t-test and found that the average acceptance level between political affiliation across all buckets was statistically significantly different ($p < 0.05$). Between Democrats and Republicans, the largest difference in opinion was whether romantic partners should have access to satellite images; 47% of Republicans found it acceptable versus 8% of Democrats. Democrats and Republicans disagreed more than men and women in what is acceptable to image (17% average disagreement versus 11%).

Policy Recommendations by Political Affiliation. For the statement questions in Section 2.6.5, shown in Table 2.6, 88% of Republicans most agreed with Statement 2,

which is to control satellite image access by price and use case. In contrast, only 65% of Democrats chose this statement.

Scenario Pair Acceptability by Political Affiliation. The largest differences in scenario acceptability were between Republicans and Democrats. Republicans were dramatically more likely than Democrats to say both scenarios are acceptable. There was almost a 50% difference in acceptability between Republicans and Democrats on scenario questions 2 and 3 (Table 2.2).

Resolution Acceptability by Gender. For assessing comfort of spatial and temporal resolutions, results followed similar patterns of women wanting more privacy (lower spatial and temporal resolutions) than men. There was not as clear of a pattern in resolution comfort between Democrats and Republicans.

Consideration 12: People's lived experiences and personal ideologies affect their privacy perceptions of satellite imagery. For example, overall, women wanted more privacy (lower spatial and temporal resolutions) than men.

Consideration 13: An average perception of acceptability is not enough; demographics that are affected most for each use case should be considered, as well as the distribution of political affiliations of citizens.

2.6.7 Study Participants' Solutions

Throughout the survey, study participants brought up possible legislation, regulation, and other limitations for satellite imagery in the free responses. One participant suggested that there should be public records of requests along with every image request. Another said there should be increased punishment of people if they use satellite imagery to commit a crime, such as stalking. With the insurance scenario, someone said it should be allowed with consent. One participant who said that satellite images should be available depending on

use case said that "satellite imagery could be super useful, but there need to be guidelines. For instance, you cannot buy satellite imagery to someone's private home" (P1). Another participant suggested image requesters must pass background checks. Others suggested location-based controls, such as preventing imaging of schools, parks, and personal property. We discuss some of these solutions further in Section 2.7.

Consideration 14: Listening to residents might yield ideas for regulatory and technical solution to prevent satellite image privacy harms.

2.7 Synthesizing Recommendations

Informed by the results in Section 4.6, we now turn to RQ2 and our synthesis of regulatory, legislative, and technological recommendations, as well as future directions. As exemplified by survey respondents in Section 2.6.2, commercial satellites offer many existing and potential benefits. It is both unhelpful and unrealistic to prohibit their use in the United States or internationally. But there are some approaches that multiple stakeholders could take to prevent or respond to possible misuse. We refer up to considerations mentioned in Section 2.6 by mentioning the Xth consideration as "**C**X". While our expository focus is on the United States, we believe that all countries and jurisdictions should consider similar issues.

2.7.1 For Regulators and Legislators

Spatial and Temporal Constraints. There is an argument to be made for most temporal and spatial resolutions. For example, legislators may decide that the resolution at which you can identify cars outweighs the potential harms of some privacy issues, but that a resolution sufficient to identify humans is too high. In 2014, U.S. legislators relaxed limits on commercial panchromatic (black and white) imagery to 25cm [97]. As pressure from the commercial sector may grow to further lower this limit, legislators should seek input from citizens when deciding on any new regulations (**C2, C12-14**). Non-resolution-based

methods for protecting privacy and sensitive information include blurring specific portions of high-resolution satellite imagery, such as individuals' faces or military bases. We also propose mosaicking multiple images together (e.g., [142]) over a region so that it does not exactly depict a specific place and moment in time as a means to protect privacy.

As mentioned in **C6**, it is not enough to just consider spatial resolution; temporal resolution should also be considered. Whether the public — anyone who purchases or views satellite imagery for something other than governmental purposes — should have access to daily, weekly, monthly, or annual imagery are all questions to consider. As imagery at more frequent time intervals is collected, making near-real-time imagery of Earth a possibility, legislators should also consider the utility of delayed release of images. This can help avoid instrumentalization of data for near-real-time combat or intelligence purposes, as suggested by geographers [15]. However, desirable exceptions to time delays and spatial/temporal resolution constraints might include emergencies, natural disasters, and search and rescue, such as locating a missing person, ship, or other entity (Section 2.6.2). In such situations, tiered user systems, as with Norway's International Climate and Forests Initiative Satellite Data Program, which makes available 3m Planet imagery available to vetted users, may make sense [129].

Other possibilities for restricting access include a Geospatial Authorization Model (GSAM), which limit access to view, zoom-in, overlay, and identification modes (e.g., [5]), or virtual data enclaves in which users access and analyze data in a virtual/remote environment as opposed to on their own local desktop [59].

Other Constraints. It is also not enough to just consider resolution constraints as if satellite imagery is in its own data silo. Geolocation data separate from satellite imagery, such as that which is gathered from people's mobile phones and sensing devices such as smartwatches, can augment the inferences that can be made about individuals from satellite imagery. The analysis of such data is sometimes referred to as "social sensing" [158], drawing parallels with remote sensing. Artificial intelligence-assisted analysis of big Earth observation and

social sensing data, a field called GeoAI, is generating increasing interest from scholars [71], particularly in China [16], where government and police agencies demonstrate high interest in new surveillance technologies. Within the security and privacy community, it is well-known that information from multiple sources can be combined to learn information not available in either source directly (e.g., [149]). Whether or not satellite imagery can resolve an individual, other markers, such as a vehicle, may be identifiable and then combined with social sensing data to profile and track someone. We recommend that legislators work with researchers to understand what additional information leakage there may be when combining satellite imagery with social sensing data.

Conditional Access. Most respondents preferred that access to satellite imagery vary by use case and cost (**C9, C10**), but there is disagreement in what constitutes beneficial or harmful applications (**C3, C4, C8**). Differences in responses among individuals and statistically significant differences in responses by demographic group (Section 2.6.6) indicate that it could be helpful for legislators to hear feedback from a diverse set of perspectives and lived experiences (**C7, C12-14**). If conditional access or blurring out only some information, like private property, is not possible, then possible privacy harms may outweigh beneficial satellite imagery use cases at certain resolutions (**C5**).

Logging Information. When satellite imagery becomes available to the public, the government could require that resolutions are below a certain threshold, and that there be a centralized, freely available repository of all non-governmental image requests as recommended by a survey participant (Section 2.6.7). As with other computer security and privacy domains, the awareness that adversarial actions may be logged could serve as a deterrent to some adversarial actions and thus could be a component in a multi-pronged approach for protecting individual privacy.

Paywalls. The state of commercial satellites today may support the idea that "we should just give up on privacy", at least in the context of remote sensing; because militaries in other

states are using it anyways, there may be no way to protect people. However, if data is free and publicly available, this minimizes costs for adversaries (**C9-11**). The barrier of image cost may mitigate some harms, such as incentives for using satellite images as a tool for burglary. For example, free information about whereabouts on social media page aided in burglaries of high-profile celebrities [89]. Some survey respondents also mentioned how they preferred a paywall (**C11**). Privacy researchers and geographers should communicate their differing viewpoints for and against making satellite imagery free and widely available. We aim to begin this conversation with this work and also with our own follow-on discussions with geographers.

2.7.2 For Companies and Designers

When developing higher spatial, temporal, and spectral resolutions in future satellite imaging technologies, both governments and industries should weigh whether these higher resolutions are necessary to achieve goals people have for their satellite imagery use case, and if that use case might itself violate individual privacy.

2.7.3 For Researchers and Privacy Advocates

Increasing Awareness and Consent. Per **C1**, people should be made more aware of the current capabilities of satellites, as well as that purchasing commercial satellite imagery is possible for individuals. If not to fix the problem of surveillance and privacy threats, knowing that an image can be taken over backyards and even at night will help people change their own expectations of privacy and possibly their behavior. Informed by the security and privacy community's general advocacy for the value of informed consent, we suggest that there be a way for people to sign up to be notified or have their property blurred out when a satellite takes images over their area (or warning them before). More research on implementation and user interest is needed.

Future Research. Our work focuses on RGB images that are perpendicular to the ground. Future work should explore the spectrum of additional privacy concerns introduced by off-nadir, SAR, and near infrared imagery (discussed in Section 2.2), as well as complementing satellite imagery data with other data (Section 2.7.1).

Researchers currently use machine learning (ML) to evaluate satellite imagery for myriad purposes, from predicting poverty [72] to evaluating shorelines [110] to detecting disease activity [120]. However, with benefits also comes possible privacy violations; it is already possible to identify cars and groups of people in satellite imagery using ML [58]. Future research could explore both the possible harms that ML brings in the context of satellite imagery, but also how to use it as a tool to automatically blur or otherwise obfuscate satellite imagery without simultaneously revealing important information, as a company accidentally did by only blurring out sensitive military locations [85]. Future work could also explore what aspects of imagery are private in what contexts, since external circumstances such as war can change what is deemed private for a given time and location. For another application of ML, the ambiguous potential of deepfakes, which may paradoxically protect individual privacy by exacerbating doubts regarding photorealistic information, should also be considered carefully.

We also encourage the security and privacy research community to consider new technical defenses. Extending the previously-discussed notion of virtual data enclaves [59], one defensive strategy might be the creation of services that answer queries about geographic locations without revealing raw imagery. More challenging, though of potential intellectual interest, is the study of ways to protect (encrypt) satellite imagery such that recipients of those protected images can perform some computations over those images (e.g., detect if an animal is present) on their own devices without simultaneously having access to the entire contents of the raw image; such a direction is akin to research focused on other forms of search over encrypted data, e.g., [3, 31, 146].

Finally, the emergent subfield in geography of Critical Remote Sensing can also be expanded to build on analysis of not only the politics and economics of the growing commercial remote sensing industry [7], but its ethics and norms, too. Comparative work

examining the regulatory and cultural contexts surrounding satellite imagery and individual privacy in other countries [16] could also prove illuminating and help identify areas of international consensus.

2.8 Summary

In this chapter, we explored privacy threats and concerns of commercial satellites in multiple phases; first, we brainstormed and qualitatively analyzed possible threats of commercial satellite imagery. Second, we surveyed people living in the U.S. about their awareness and concerns towards satellite imagery. We found that there was insufficient awareness about commercial satellites and that people's privacy expectations and preferences often conflicted with current commercial satellite practices, and may conflict even more with future commercial satellite practices. Third, we proposed possible mitigations of potential privacy violations addressed to multiple stakeholders. Although not considered later in this dissertation, these mitigations will hopefully serve as discussion points among commercial satellite imagery stakeholders. High-resolution commercial satellites may change how we understand surveillance and our expectations of privacy. Before spatial and temporal resolution improve even more, it is crucial to study these topics and begin holding conversations with legislators, industry, and the public that is being watched from above.

Chapter 3

Security of Small and Amateur Satellites

This chapter looks at another element of the satellite ecosystem, this time from a security perspective. In particular, small satellites run by clubs. The democratization of space through decreased hardware and launch costs has enabled students to build and launch their own satellites. While this may seem harmless currently, the increasing capabilities of many of these club satellites, such as propulsion, make them a possible target for use in future space cyber warfare. Compromising university smallsats could be used to target other assets in space or create excessive amounts of debris through their destruction, which can then collide into other space objects.

Through a combination of interviews and code audits, this chapter identifies barriers to building small satellites securely, poses directions to resolve some of these security barriers, and brings attention to emerging threats to the satellite ecosystem. A shortened version of this chapter initially appeared as a paper titled "Unencrypted Flying Objects: Security Lessons from University Small Satellite Developers and Their Code" at the Usenix VehicleSec workshop in 2025 [111].

3.1 Motivation

Satellites have increased in popularity and magnitude in the past decade [1]. Satellites face a multitude of security challenges that differ from hardware on the ground today, such as the impossibility of physical access post-launch and limited contact periods as they orbit the earth. The decreasing cost of launching satellites [4] has enabled more small players to launch, including amateurs and university teams. These amateur university teams typically operate smaller satellites, called "smallsats." Academic smallsat use cases include testing

university researchers' scientific payloads and giving students hands-on learning experiences. Despite being small projects and not-for-profit, university smallsats still face risks in space. One oft-cited risk is the Kessler Syndrome [80,81], where debris caused by satellites colliding with other objects could grow exponentially after reaching a particular threshold and thereby make future space exploration hazardous. Unsecured amateur satellites could harm the ability for emergency response given that amateur satellites can aide in this process [93]. Or smallsats might be used as "kinetic weapons", as discussed by interview participants and by Kurzrok et al. [88]. Smallsats could even incite international incidents or cause physical harm to humans, as we elaborate on in our results. Therefore, for the space ecosystem to be safe overall, we argue that amateur smallsats must also be secure.

Research Questions To gain insights into computer security practices with smallsats, and with our focus on university satellite clubs as a window into the broader space, our first research question is:

- RQ1: What are the security goals and practices of university satellite teams?

To understand whether and how to improve smallsat team security, we first need to understand existing goals and security practices. We analyzed this first through interviews with the leaders of 4 satellite clubs [1] (8 interviews in total, an average of 2 leader interviews per club) since club leaderships' perspectives on security affects the entire club's priorities. The aim was to surface concerns and practices across different teams rather than to make claims about generalizability. Thus, we chose qualitative methods and consistent analytical choices designed to summarize and describe our data. In other words, our research intends to uncover a richer and more nuanced understanding of the practices within this particular sample, rather than to make inferences about generalizability to other samples [17].

Second, we performed an audit over dozens of repositories from 3 of the teams we interviewed, parsing through code, design diagrams, and documentation. We focused on

[1] 5 clubs met our search criteria; 1 club did not respond to inquiries for research, resulting in interviewing 4 clubs overall

understanding the high-level security designs and choices of each club and club member, and comparing the designs in the code to their stated threat models in the interview; this type of design-level reconstruction and analysis on bespoke systems is not possible with automated tooling, therefore our analysis was manual. We detail our approach in Section 3.6. To foreshadow our findings, security practices varied widely across teams, with none of the code we analyzed meeting all security measures needed to prevent malicious actors from sending commands up to the satellite while it is in orbit.

- RQ2: What barriers to building satellites securely do smallsat teams face?

We identify a variety of barriers to building satellites securely. Club members weigh tradeoffs between other goals (e.g., launching on time) and security. We also identify gaps in security knowledge, threat modeling, and implementation that serve as barriers to secure satellite implementations. For example, we find that security goals/non-goals vary greatly not only between clubs but *among* club members, highlighting a non-uniform approach to security. Understanding these barriers can help us recommend solutions.

- RQ3: What processes and tools can we recommend to improve smallsat team security?

Through a combination of interviews and our own knowledge as security researchers (4 of the authors), members of a university satellite club (2 of the 4), and as a leader within a satellite club (1 of the 4), we identify design recommendations and tools. These recommendations aim to bridge the gap between current satellite security practices and a more thorough security posture for smallsats appropriate to the university satellite context, i.e., groups with limited time, budget, and security expertise. We anticipate that some of these recommendations will apply to industry, as multiple participants indicated limited security considerations in industry. Additionally, many satellite club members end up working in the space industry (as we observed in our sample of leaders and our own satellite club experience), so understanding and addressing the security barriers at the university level may equip future space industry professionals with tools to build more secure industry space vehicles.

Emerging Technology Security Research It is well-known in the field of computer security research that computer security often lags technology innovations. For example, the modern automobile became computerized long before the automotive industry embraced computer security as a practice [86]; likewise, the medical device industry incorporated computation and wireless communications before embracing computer security [63]. The goal of our work is to contribute to the study of the emerging smallsat computer security domain — a domain that, like automobiles and medical devices before, is exploding in innovation [1, 4] but, to our knowledge, has not yet ubiquitously embraced the need for strong and universal computer security practices.

Whether today's smallsats are presently at risk of exploitation is speculative, but hacking enthusiasts have already demonstrated the ability to compromise government satellites [105], and commercial satellite communications networks have also been targeted [106]. We return to assessing possible present and future risks in Section 3.5. Studying university smallsats not only helps us explore the challenges and risks that these specific non-profit smallsats face, but given the proprietary nature of many smallsat manufacturers [43, 103, 163] these open-source smallsat groups may be the closest possible window into satellite security overall.

3.2 Background and Related Work

3.2.1 Satellite Applications and Overview

Though government-owned and commercial satellites are the most widespread today, hundreds of university satellites have launched since the 1980s, with launches increasing each decade [148]. Even high school and middle school students have built and launched satellites [52, 138].

University satellites support educational goals such as testing scientific payloads, hands-on learning experiences for students, and collecting space or Earth data. Some also serve as communication relays for amateur radio satellites (*amsats*). Amsats facilitate ham radio communications, which can sometimes aid emergency response [93]. These

emergency response utilities are another reason we believe that amateur smallsat security should be further studied. University teams often use amateur radio frequency bands for communication with their satellite. While ham radio usually requires communications to be unencrypted, it allows satellite command encryption [48].

Amsats are often supported by open-source infrastructure and volunteers who collect and relay telemetry using their own ground stations. One such example of this infrastructure is SatNOGS, an open-source ground station network [139]. Multiple clubs we interviewed used SatNOGS to get data from their satellites from parts of the earth not accessible with their own ground stations. Many satellite clubs have open-source code repositories that are publicly accessible before and after their launch, which we discuss further in Sections 3.5 and 3.6. Another common resource is the the suite of communication protocols established by the Consultative Committee for Space Data Systems (CCSDS) [23]. These protocol specifications cover a number of protocols across the network stack, some but not all of which include security measures in their specs.

3.2.2 Satellite Architecture

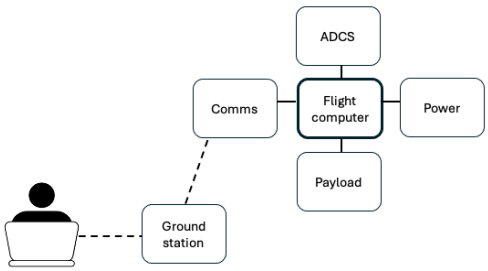


Figure 3.1: Overview of components in a generic smallsat. Ground station communicates with the satellite radio (comms). The flight computer communicates with all subcomponents, including power, ADCS, the satellite's payload, and comms.

At a high level, a smallsat consists of a few subcomponents, shown in Figure 3.1:

Ground station The ground station is composed of a radio, antenna, and computer to send commands up to a satellite. A satellite might only receive commands from a single ground station, or a network of ground stations in different locations could help increase contact time with the satellite. The portion of communication sent from the ground station to the satellite is called the *uplink*, and the inverse is called the *downlink*.

Communications space segment (Comms) The comms segment on the satellite receives commands from the ground station via the uplink, and sends back telemetry (data about the satellite's state) and other command responses via the downlink. Some satellites have the ability to communicate directly with other satellites, especially if part of a fleet of satellites, but this is less common with university satellites.

Flight computer The flight computer schedules and facilitates communication between each subcomponent.

Attitude determination and control system (ADCS) ADCS helps control the orientation of the satellite, which is critical for keeping the satellite stable and ensuring that the satellite's solar panels are pointed toward the sun. Some satellites use propellant to maneuver, while others just use magnetorquers and reaction wheels.

Payload Payloads are the "purpose" of the mission. Examples include cameras and novel batteries to test in space.

3.2.3 Satellite Security Challenges

Securing satellites poses difficult security and reliability challenges, including:

Contact constraints In low Earth orbit, satellites take around 90 minutes to orbit Earth. Single ground stations can typically only contact a satellite for 7-10 of those minutes [39]. And if the satellite were to enter an unrecoverable state, there is no way to physically access the satellite to reboot it.

Latency constraints Satellites may need to respond real-time for tasks like collision-avoidance or ADCS [75]. Adding encryption and other security measures could increase the amount and size of packets needed when communicating. With the very limited bandwidth

that many smallsats have, added packet size/count could lead to slower communications, ore even violate real-time constraints.

Processing constraints To meet their size and weight constraints for launch, smallsats typically have a limited power supply. Due to this, they have minimal processing power. Every processing step, including encryption and decryption, takes up valuable power and compute time [103].

Custom Systems Many satellites are largely bespoke systems, with custom protocols, operating systems, and hardware components. This makes it difficult to use existing security tools, such as fuzzing, as supported by Willbold et al. [162]. Our security analysis in Section 3.6 was therefore manual.

3.2.4 Satellite Security Research

A wide body of academic work has enumerated and taxonomized security and privacy threats of and to satellites [32, 43, 45, 73, 83, 84, 103, 112, 126]. Pavur et al. compiled a set of satellite threats enumerated in previous work and explained the history of satellite security incidents by decade [126]. This threat modeling aided in our own threat modeling in Section 3.6.

While many papers have focused on threat enumeration, increasingly work has taken a more experimental approach [57, 74, 75, 91, 107, 156, 163]. Willbold et al. [163] analyzed firmware security of various satellites and surveyed industry members about security practices, highlighting a need for improved security among industry satellites. We build upon this work by more deeply exploring the university satellite security space, surfacing unique security barriers and needs.

Some work has highlighted the need for more security on university and other smallsats [79, 88, 137]. Kurzrok et al. explored the theoretical extent that a hacked smallsat could travel under different propulsion methods [88], showing the physical danger of unsecured ADCS telecommands. Kurzrok et al. also mentioned informal conversations among the smallsat community suggesting university teams often do not encrypt their telecommands and highlighted the need for a systematic analysis on university satellite

Table 3.1: Interview participants by identifier. Security knowledge ranged from 1 (novice) to 5 (expert). Knowledge was based on self-reported scores except for D1. A * next to their score means that the respondent did not give a specific number, and therefore the score was an estimate based on their interview responses.

ID	Club	Industry Experience	Security Knowledge
A1	A	✓	3
A2	A	×	4
B1	B	✓	3
B2	B	×	3
C1	C	✓	2
D1	D	×	1*
D2	D	✓	2
D3	D	✓	5

security. Our work intends to fill this gap.

3.3 Methodology

We used a combination of interviews and code analysis to explore the security practices of university smallsat teams. Our methodology involved five steps: 1) identifying university clubs online, 2) selecting clubs to interview and analyze based on publicly available code, 3) assessing the code to identify code-specific interview questions, 4) conducting interviews, and 5) analyzing the security of the code, cross-referenced with interview responses, based on our satellite-specific threat model. We interviewed 8 participants across 4 clubs and looked through all publicly available repositories for 3 of those clubs.

3.3.1 Finding Clubs to Analyze

We used a search engine to identify as many university satellite clubs as possible, resulting in 41 overall. We then filtered for clubs with public code repositories and documentation. This resulted in 7 clubs, 5 of which are in the U.S. We chose public code repositories 1) because we did not wish to pressure clubs into revealing their code that could then lead to the possibility of them becoming de-anonymized, 2) avoiding conflict with their funders or institutions who might require that the code remain private, and 3) preventing violation of any ITAR guidelines, since many satellites have ITAR restrictions where certain implementation details cannot be shared to non-U.S. citizens. We chose not to interview members from the 2 clubs outside the U.S. because our research team is within the U.S. and regulations and policies may vary by country. Focusing on clubs within the U.S. allowed us to minimize variables including regulations and local educational cultures. Further, 1 of the 5 U.S.-based clubs did not respond to requests for interview and thus is not included in analyses. Finally, we interviewed Club D but excluded a code analysis after determining the available repositories corresponded to previous satellites our interviewees had not worked on (whereas more current code was not yet publicly available). We determined this only after Club D interviews started, so we still report their interview responses in Section 3.5. While this may be a small set of clubs, there are not many clubs in the first place that have actually launched satellites per our searches, and a smaller set of those that have their code open source.

3.3.2 Interviews

Our interviews take a standard approach by surveying developers to understand how security and privacy are treated in practice [133, 135, 165]. This approach allows us to understand gaps between research and practice and pose solutions towards reducing these gaps. We interviewed participants from 4 clubs.

Recruitment and Screening

To help answer RQ1, we especially recruited for members who worked on (and preferably led) satellite software. We also recruited satellite club leaders because their preferences likely affect satellite security implementation downstream. We found contact information through websites and public documentation. We reached out by email to each club and relevant club leaders, members, and alumni. Table 3.1 summarizes our interview participants.

Developing Interview Questions

We constructed a set of interview questions based on our research questions. The first author constructed a semi-structured interview script, with pre-determined questions but allowing for flexibility (e.g., omitting questions, rewording, or asking follow-up questions as necessary depending on an interviewee's responses). This interview script was evaluated and revised by the authors and an external security expert. Two authors piloted the interview script with another author and an external colleague in robotics.

Interview Procedure

Interview questions below are labeled with their corresponding research questions.

Two authors led the interviews virtually over Zoom meetings. The first author asked questions from the script while the second interviewer took notes, and both asked follow-up questions as necessary. We did not mention security until the security-specific questions to avoid biasing participants' responses about their goals and concerns when building a satellite [56, 92, 136]. The interview script is included in Appendix B.1.

General Questions. To get a better understanding of participants' club experience, we asked about the satellites they worked on, what years they were in the club, and whether they had worked on satellites outside of the club.

Mental Models. We asked participants to draw a high-level diagram of how their satellites worked, and asked those with software experience to focus most on the software

components. We used this diagram to help the participants and us as interviewers visualize further potential security threats. We did not yet mention security in this or the previous sections. We asked what goals they have when building satellites (Research Question 1) and what scenarios they were concerned about. These scenarios revealed whether security issues were a concern at the top of their mind (RQ1). We also asked about goals to further understand what tradeoffs to security they might be willing to make, which may serve as a barrier for implementing security (RQ2). Some of the wording and questions in this section and the following security-specific questions were inspired by Zeng et al.'s exploration of security and privacy perceptions of IoT users [169].

Security-specific Questions. We next asked whether they had any security-related concerns, or if they had other security concerns that they hadn't mentioned yet (RQ1, RQ2). We asked what they think the worst thing that could happen is if the satellite is not secure, who might attack a satellite, what discussions the club had about security, and how much they discussed security compared to reliability (RQ1, RQ2). These questions assessed participants' overall threat models by asking implicitly about assets (worst-case outcomes), adversaries (who might target the satellite), and vulnerabilities (what their security concerns were). We then asked what security tools they already used (RQ1) and what they wish existed (RQ3).

Demographics. We asked about the participant's area of study, satellite team size and budget, their satellite industry experience, and their cybersecurity knowledge.

Closing Questions. We asked if there was anything else the participants wanted to share and whether they expected us to ask anything else, a recommended practice in HCI research [92]. We also asked several participants code-related questions relevant to their club's repositories to gain further insight into their procedures.

Data Analysis

For each interview transcript, both interviewers used emergent coding [92] to note concepts that arose in each transcript. We compared themes that came up, then used axial coding to

categorize these themes into groups. We translated these axial codes into a final codebook (Appendix B.2). For the frequency results in this chapter, both interviewers coded based on the videos and the transcript. We coded these frequencies independently and calculated initial agreement using Cohen's Kappa, with substantial agreement between coders ($\kappa = 0.75$, $SE = 0.08$). We then further refined our codes and discussed differences in coding to reach agreement.

3.3.3 Software Analysis

Our analysis of the clubs' satellite software was split into two parts. First, two authors identified and analyzed repositories (along with public documentation and diagrams) belonging to each satellite club. As mentioned in the introduction, these analyses were entirely manual due to the goal of analyzing whether clubs implemented the high-level security goals that they stated in their interview, as well as the inherent difficulty in running automated tooling on disparate embedded platforms [2, 162]. From these analyses, we then derived high-level block diagrams (like Figure 3.2) of each satellite's architecture and system functionality. Second, four authors (all security and privacy experts) synthesized a threat model for each evaluated satellite based on these high-level diagrams. If our threat model differed from the threat model of the participants (as expressed in the interviews and corroborated by the software), we evaluated whether the code met the security requirements of *our* threat model. We argue that this code analysis methodology serves as both a barometer on if the clubs' perceptions of their security postures are *accurate* (i.e., the satellites behave how their clubs expect them to based on their mental models of the satellites, other interview responses, and comments in their code), as well as an indication of threat model *suitability* (i.e., whether the threat models are appropriate for the satellite's goals, as judged by security and privacy experts).

Our manual code analysis was primarily guided by two authors' experience with satellite software. After identifying the clubs' Github organization and associated repositories, we filtered for security-relevant repositories. Since many of the repositories were for hardware

designs or design collateral (Appendix B.3.1), the most relevant repositories tended to include software for ground stations, communication systems, and flight computers. For each of these repositories, we read the code at a high level to understand its intent and how it fits together with other subsystems. Rarely, we were able to additionally execute code that did not depend on bespoke embedded hardware (such as for Satellite C). Based on these insights, we made diagrams for each satellite that describe (at a high level) how the security-critical components interact.

We then specified a general threat model (Section 3.6.1) to guide a more in-depth security analysis for each satellite. This threat model defines a remote adversary who can reverse-engineer the satellite's communication protocol and send arbitrary commands to the satellite. Our in-depth code analyses therefore typically began at each satellite's communication subsystem, specifically the handling function for received messages. We explored the threat surface of available commands, noting any reachable commands that might violate our participants' threat models, or confirm/deny any of their perceived risks or goals (Section 4). If the participants noted that they implemented encryption or authentication, we verified whether this was the case. Finally, we also traced backwards from the participants' security and safety goals, and checked whether any internal state could lead to a violation of these.

3.3.4 Scope and Limitations

Our goal is to understand the computer security practices, barriers, and opportunities within the small yet growing community of university satellite clubs and their members. Our sample size was limited, in part because we recruited from a small and specific population (on the order of dozens of clubs based on our searches), which is itself a valuable endeavor [17] and aligns with our broader research goals. Though our goal is not generalizability, qualitative methods can afford transferability [33], such that lessons and recommendations uncovered here may be valuable in understanding other university satellite teams. Further, security vulnerabilities in only a few launched smallsats still have the capacity to cause harm to the entire space ecosystem, which is why it is so important to study this overlooked group

of satellite operators. All participants interviewed were subsystem leads or heads of their clubs, so our results speak to club leaderships' perspectives on computer security and privacy. While there might be members within a club with different security knowledge than those that we interviewed, their existence does not detract from our findings with club leadership. Nevertheless, like any interview study, the results may suffer from self-reporting bias. We attempted to mitigate this through cross-referencing discussions of security with our manual code audit (except for Club D).

In terms of code analysis, it is possible that we missed relevant vulnerabilities given the scope of the repositories and the difficulty of using any automated tools on heterogenous, bespoke embedded systems. Because our security analysis focuses on the design-level, we attempted to mitigate this by anchoring our code analysis on the satellite-specific threat model we developed based on each payload, rather than analyzing for as many potential vulnerabilities as possible.

3.3.5 Our Perspective

Our team of authors has experience with both attack-focused and defensive research. Two of our authors are members of a university satellite club and have experience developing components for a satellite. One author has experience as the lead of multiple subsystems within a satellite club. Three authors have expertise in participant-focused studies, including one author who is a social psychologist. Two authors have amateur radio communications experience. Multiple authors have embedded systems hardware and software expertise.

3.4 Ethical Considerations

Our institution's IRB approved the interview study under the exempt research category. Before each interview, we shared a consent form that participants agreed to verbally. Participants optionally consented to video recording through the Zoom recording functionality so that we could have an accurate script and record any drawn diagrams.

All participants agreed to the recording option. One interviewer manually transcribed participant responses during each interview.

We paid each interview participant $75 for their responses with an expected time of 1 hour per interview. This pay rate is higher than usual for interview studies; we wanted our pay to reflect an estimate of the time it would take out of an industry job to participate in our interview, since some of our participants were in industry. All questions were clarified to be optional. We anonymized study participants; the goal of this study is not to focus on individual clubs, but rather to understand what the range of security practices, preferences, and security barriers might be for university and other smallsat teams. To further protect participants, we sometimes obfuscate which anonymous participant said what in the combined results when we believe that it would not affect the interpretation of the response. For security vulnerabilities that we found that satellite clubs were not already aware of, we disclosed our findings to them. We offered our help to answer any questions or help resolve the vulnerabilities if needed.

3.5 Interview Results

We present below our interview results combined across clubs. Based on these results, we note strengths of security practices among teams, as well as specific barriers that might lead to difficulty implementing appropriate security measures. These results then inform our discussion of potential tools to help smallsat teams build more securely (Section 3.7).

3.5.1 Participants

We interviewed 8 former and current university satellite club members about their satellite club experience across 4 clubs. See Table 3.1 for a summary of participants. Participants consisted of club leaders, subsystem leads, software leads, or a combination of the above. 4 members were current and 4 were former members, graduating 2018 at earliest. 5 out of 8 members had experience in the space industry, both through internship and full-time roles.

Security knowledge ranged from 1 (novice) to 5 (expert). Interviews lasted between 30 and 75 minutes.

3.5.2 Satellite Goals

To understand what tensions between security and other factors might exist, we asked participants about what overall goals they had when building their satellites. As a reminder, we did not mention security prior to the security-specific questions later in the interview. Answers fit broadly into the following categories: scientific missions, educating club members, educating community and other clubs, building the satellite cheaply, and building the satellite reliably.

Clubs' scientific missions included taking images of Earth, testing battery systems in space, and relaying sensor data from remote regions. Clubs prioritized hands-on satellite development to educate club members, while outreach activities and payloads were often used to educate communities.

Some clubs tried to build cheaply to stay within their own budget, while others also tried to build cheaply so that other clubs could replicate their designs in the future on a limited budget. They did this by relying minimally on off-the-shelf components, instead opting to build their own components from scratch or from open-source software and hardware. One such example was using the (unencrypted-by-default) OpenLST transceiver, a component with high manual cost to integrate but costing under $100 per radio.

Barrier 1: Low-budget and education goals can lead to using homebrewed designs, which might influence the security tools used.

3.5.3 Difficulties faced

Non-technical difficulties were emphasized the most. Participants discussed a wide range of difficulties within the club, both technical and non-technical. One of the most common categories of difficulties was non-technical issues relating to lack of various

resources, with 7 participants bringing up difficulties in this category. Resource deficits included insufficient time for development, lack of people, limited experience, insufficient documentation within the club to transfer knowledge, insufficient public documentation to help clubs figure out what and how to build, difficulty finding materials and materials being discontinued, limited budget, and limited access to expensive physical testing infrastructure.

Technical Difficulties Varied. Technical difficulties mentioned varied by club and depended on both scientific missions and hardware used, though creating good software testing infrastructure and simulations was mentioned as a difficulty by 4 participants. Participant A2 provided a concrete example of this when they happened to find a memory corruption error:

> While our attack surface was pretty limited, the worst case I suppose would be someone managing to gain remote code execution via a malicious uplink command entry point... I discovered an array out of bounds issue at one point while testing myself when I noticed some data was corrupted and fixed it; we didn't have good automated tests to catch that kind of stuff even at the beginning. Most microcontroller code is written in C which is notorious for letting you shoot yourself in the foot with memory issues... That combined with amateur software engineers is a potentially dangerous combo, especially for missions where a malicious actor could do more damage. (A2)

Barrier 2: Insufficient testing — potentially due to lack of time and other resources — can lead to missing important satellite security flaws.

3.5.4 Security Discussions During Development

The remaining portion of our interview focused on security. Only 1 participant mentioned insufficient security as a potential scenario of concern in our interviews before we explicitly

asked about security concerns. 6 participants across 3 clubs said that security was discussed in the club once we asked.

Security perceptions differed *across* and *within* individual clubs. Whereas some clubs did not authenticate or encrypt their commands, others did. However, the differences in security practices did not correspond with actual threats. For example, Satellite B had a propulsion system but did not have encryption, while D3 said they would implement encryption even though all participants from Club D felt that there was minimal risk for their satellite.

There was also disagreement *within* individual clubs about whether a security protection was employed (Club A, Club B) and whether security was discussed at all (Club D). We discuss this more in the code analysis section of individual clubs in Section 3.6. D3 said that reliability "is mainly how I sold my [security] research and thesis." Their research was mainly on fuzzing, so this meant that finding security issues, such as memory corruption, was important for reliability as well. However the Club D leader said that there was no security discussion or security concerns with their satellite.

Barrier 3: Inconsistent security goals and discussions within a club can lead to non-uniform implementation of security across the entire satellite system. One subsystem or code segment might have considered security while others have not.

Some clubs employed aspects of threat modeling in their design phase. Both Club A participants said that security was discussed. A2 mentioned that their team met to whiteboard all the things that could go wrong. One design point they discussed in whiteboarding sessions was whether or not to allow software updates. On the plus side, being able to fix a bug after launching could save the mission. The downside is that this is complex and could introduce additional risk if incorrectly implemented. Thus, they decided not to include a software update mechanism. While software updates were discussed, the participant did not explicitly mention their value from a security perspective. Another

example of at least partial threat modeling is participant D3's security fuzzing project; they mentioned using fuzzing to find bugs that would prevent the satellite being taken over and having firmware flashed onto it.

Strength 1: Some clubs practice threat modeling in their satellite development process.

3.5.5 Satellite Threat Models

Participants discussed negative outcomes but mostly thought they were unlikely. Participants brought up a wide array of potential threats to their satellite. 6 believed that these outcomes were unlikely, which often affected their security practices. For example, D1 believed that the satellite's sensor data was not sensitive and thus would not be a target.

We do not wish to imply that their risk assessments are misguided; based on our interview discussions, we have no evidence to suggest any club that has already launched their satellite was targeted by attackers. However, as threats change in the future and as the capabilities of university satellites change over time, these threat models may not hold. We see theories of this changing threat landscape, including space warfare, further into these results.

Both club members and third parties had concerns. For example, participants said that NASA was concerned that the lights on Club B's satellite could harm astronauts. The participants framed the blinding concerns as a safety issue rather than security issue.

B1 recalled that some outside groups, though they could not remember who, required Club B to think about security. B2 as club leader needed to ensure that all systems complied with International Traffic in Arms Regulations (ITAR [123]), which sets some controls over how advanced various technologies on the satellite can be while still being open source, such as the propulsion and star tracking system. ITAR is enforced for national security and safety reasons rather than preventing the satellite itself from being exploited.

Of Note 1: In some cases, security considerations might have been imposed from the outside, though the reasons may have been for safety.

Some threats were discussed during the design process while others were only discussed in the interviews. Throughout our questions about security, club members mentioned a variety of possible ways to attack a satellite and potential security issues. These issues include sending up malicious commands, exploiting memory corruption vulnerabilities, performing remote code execution, denying service, unintentionally using malicious third-party code, and physical access pre-launch allowing someone to upload malicious code onto the satellite. However, some concerns only came up in the interviews and not while building the satellite. Remote code execution, third-party code, or physical access pre-launch were not discussed amongst club members while building the satellites. Some participants had industry experience where they were required to think more about security, which may have helped them with their threat modeling during the interview. The limited discussions of security within clubs despite many current members being aware of other potential security issues corroborates that security was not a main priority when building the satellites. We discuss what tools were used to mitigate these risks in Section 3.5.6.

Barrier 4: A lack of extensive threat modeling prior to launch means that some attack vectors were not considered.

Worst-case outcomes. For satellites that had propellant, one outcome that participants mentioned included sending malicious commands to ADCS to wreck their satellite, causing space debris. Even worse, B2 mentioned how the satellite might hit another satellite or the International Space Station (ISS). Some participants who worked on satellites without propellant or believed their propellant was not sufficiently powerful were not as concerned, with 1 participant mentioning that hackers could only retrieve public sensor data, and 2 others saying that they could lose communication with the satellite or otherwise

be denied service. The above concerns were only considered during the interview, not prior to launch.

Physically harming astronauts Club B had an extremely bright flashing light on their satellite. Club B participants mentioned blinding astronauts as a key concern in their design phase since their satellite was planned to launch from the ISS and would thus start out next to astronauts. Section 3.6.2 shows what preventative measures they took to mitigate this risk. Club B participants mentioned that third-party providers who were helping launch the satellite were also concerned of this risk. This physical safety concern is the only threat considered prior to launch in this section of worst-case outcomes.

Violating Regulations C1 mentioned that attackers could use satellites to violate regulation. For example, they could violate FCC guidelines and broadcast messages when they are not supposed to, or use a satellite's on-board camera to violate regulations and take images of other satellites rather than images of Earth [65].

Inciting International Incidents Another concern that came up was that some foreign government could take hold of a university satellite from the U.S., use it to cause harm, and then blame it on the U.S. to limit U.S. presence in space, or to incite an incident. This concern came up without any prompting other than the interview script, and it shows an example of what the future of space warfare might look like:

> The worst case that I can think of is that a foreign adversary takes control of the satellite and [causes it] to run into the International Space Station... [or] crash into some sort of satellite that's part of an international collaboration. At that point, the amount of paperwork that you're going to have would build a mountain high enough to get into orbit... Any time that you have a satellite with a propulsion capability..., you effectively have a kinetic weapon that can destroy other satellites... A traditional adversary to the United States could take control of some U.S. satellite and say "The student satellite is malfunctioning and it has crashed into our satellite ... This is a big international incident ... The United

States can't control their satellites." (B2)

This threat implies that any satellite with sufficient propulsion systems should consider its security, even if it's a university smallsat. Thus, teams should be aware about how this kind of threat might become more likely in the future as more warfare involves space [116].

Barrier 5: Incomplete threat models about why an adversary may target a club in the future (for example, to incite an international incident) can lead to differences between real threats and implemented threat mitigations.

Potential Adversaries. When we asked from whom the clubs' satellites might need protection, answers ranged from those hacking out of curiosity and boredom, to foreign government adversaries. Again, there was no indication that most of these adversaries or the adversaries below were considered prior to the interview. Some participants explicitly mentioned adversaries that they thought would not target their satellites, including foreign governments. Some explicitly mentioned intentions that adversaries would *not* have towards university satellites, including not wanting money and that there would be no national security benefit to hacking them.

Environmental Advocacy Groups Some participants based potential adversaries off their experience. Club B received a letter from an environmental advocacy group against light pollution asking them to stop development of the satellite, and proposed that this group could be a potential adversary.

"Space Pirates" Some participants posed that future space warfare might include "satellite pirates" or "space pirates". While this does have an existing definition regarding piracy of satellite-distributed content like satellite television, D2 had heard of satellite pirates as for-fun hackers who try to decode messages and use it against people. D3 heard of "space pirates" through previous work on a government-funded satellite security project. D3 defined space pirates as a satellite hacker that will become more important as space warfare becomes more prevalent [116]. D3 said that space pirates could even refer to someone with physical

access to the satellite in the future: "someone could have a spacecraft where they can approach your satellite or grab it out of space and modify its hardware."

Barrier 6: Conducting an informed risk analysis of adversarial threats in space may be challenging, since discussion of some risks and predictions of risk, like "space pirates", require information not readily accessible in the open literature and therefore not as accessible to university satellite clubs.

3.5.6 Security Strategies and Tools Used

Clubs used a variety of tools and techniques, both effectively and ineffectively to try to implement security. In addition to whiteboarding out worst-case scenarios (Club A), participants said that they did manual security audits of code, are building fuzzers, and doing "hardware debugging" (Club D). As we will see in Section 3.6, some security protections employed were partially incorrect. 4 participants did not know what security tools were used or said that none were used. One tool for encryption mentioned was "frequency modulation," so there was likely confusion about the term encryption. Based on participant ideas and our own analysis, we discuss potential future tools to help improve security in Section 3.7. Another security choice was to "scrub commands" from the public code (Club B). The command scrubbing in Club B was not effective for preventing adversaries from knowing the commands; see Section 3.6.2 for more details.

Barrier 7: Clubs may not be currently equipped with sufficient security tools and/or understanding of the security tools to effectively secure their satellite.

3.5.7 Security Tradeoffs

All clubs made tradeoffs that directly impacted security, as any organization must do. Some clubs chose to forego confidentiality since they intended the communications downlink (satellite sending info to ground station) to be public for educational purposes, or were

required for it to be public if they were within an amateur radio frequency band. One participant believed they could not encrypt the commands to the satellite in the uplink due to amateur radio laws, but this is not the case. Based on the authors' own experience in satellites clubs, this misconception is not uncommon.

Barrier 8: Misunderstanding in regulations might lead to implementing less security protections.

Club A chose not to encrypt their uplink because they were concerned that they might implement it incorrectly and had seen other clubs "shoot themselves in the foot" when trying to implement encryption (A2). Club A decided it was a reasonable tradeoff because they did not think there was enough of a threat to justify encryption. Another participant said they cared a lot about maintaining simplicity of their software, which in turn would help with maintaining reliability (D2).

Barrier 9: If a security solution has potential real or perceived negative reliability impacts, there may be resistance to the adoption of the security solution.

3.5.8 Security Issues in Industry

5 out of 8 participants had industry experience specific to satellites or in the space industry overall. 4 out of 5 only had industry experience after leaving the club, while 1 had industry experience during undergrad through internships. 4 participants said that industry takes security more seriously than their university satellite team (though one of these participants works now on rockets and not satellites). 2 participants said that there were insufficient security practices in industry. Some responses show a lack of priority in security in government satellites and possibly beyond:

> Any internships that I've applied for that have involved testing ... has been mainly on a reliability basis. (D3)

> Not my specific company, but the space industry as a whole is hilariously bad at cybersecurity. It is comical and also terrifying, with the scope of projects in industry that have absolutely zero encryption. They're waking up to that now . . . and it is being taken more seriously. . . Cyber warfare has become much more of a concern in general (C1)

D3 was going to be involved in developing the security of a government-funded satellite communications project that their university was involved in, until the funding for the security initiative was cut (at least for D3's institution):

> We were involved with [the satellite project] in the very early phases. . . I think our group was chosen to do the security portion and then a month later when they were getting the cybersecurity budget sorted out for this project, they decided the budget would be $0. (D3)

D3 worked on another satellite project outside of Club D:

> I was in integration testing on [a government satellite communications system]. As far as I could tell. . . , they did no unit testing, no fuzzing, no security auditing. All the testing for catastrophic failures was done by people. . . I was there for one or two potentially disastrous problems with the flight software that almost ended the project. I think most security stuff is left up to chance and most of these projects trust encryption as their only way to prevent hacks. (D3)

Of Note 2: Security practices of satellites in industry and government may be insufficient relative to the satellites' capabilities.

3.6 Code Analysis

We now report on a code analysis of 3 clubs' satellite implementations, and compare and contrast these with security perceptions from our interviews. This code analysis encompasses

dozens of repositories and many thousands of lines of code; Appendix B.3 lists statistics about these repositories.

3.6.1 Threat model

To inform our code analysis and satellite-specific threat models, we first establish a more general threat model. We assume an adversary on the ground who only has access to public documentation about the satellite, i.e., is not an insider to the club. Public documentation includes all open-source code repositories, documentation within those repositories, or any other information about the satellite available on the internet. Public documentation also includes network traffic, since adversaries could reconstruct the protocol based on observing commands sent from the ground station [19]. The adversary can construct their own ground station or use a ground station as a service company and send commands up with a software-defined radio or other tool.

We intentionally exclude pre-launch physical access to satellites and their underlying hardware by the adversaries from our analysis; therefore, supply-chain attacks and physical access are not part of our threat models. We are *not* saying that these two threats could not compromise the operation of a university satellite. Rather, given that building satellites to handle supply chain adversaries is an emerging area of research [75], we do not expect (and did not see) any defenses for these threats in the code that we analyze, and hence we define it as out of scope in our analysis. Second, all satellites analyzed were vulnerable to a much less privileged remote adversary, making whether the satellite could defend against supply chain attacks not relevant. To clarify, excluding these results does not impact our interview results, as the participants were not told to consider a certain threat model. Still, we explicitly acknowledge that future works should consider supply-chain attacks, especially as satellites become more ubiquitous and capable.

3.6.2 Satellite A

Satellite A launched in 2018 and decayed from orbit in 2020. Satellite A had two main science missions: 1) to test a battery technology in space and 2) to implement a "flashsat" using a 40000 lumen LED panel, designed to be seen from space by the naked eye. For the rest of this section, we use the term "flash" to refer to the LED panel, rather than persistent storage or the process of installing software updates into that storage.

We find that the Satellite A designers were primarily focused on mitigating flash panel safety concerns, which they did successfully via precise state transitions (Figure 3.2). However, we identify availability and confidentiality issues.

System Overview

Bootloader The command/control/communications (C3, aka flight computer) software component consists of a low-level bootloader and the main real-time OS (RTOS). The bootloader is a tiny program which runs when the C3 first starts up. Its main task is to interface with an external MRAM (*m*agnetoresistive *r*andom *a*ccess *m*emory) chip to load the larger operating system image. MRAM is inherently tolerant of radiation (i.e., not prone to bit flips), and is therefore a safe place to store the large, mission-critical operating system.

Operating System The main OS is built on FreeRTOS 9.0.0, a widely-used RTOS with a broad range of hardware support. The OS performs low-level hardware initialization, bootstraps the scheduler and other core subsystems, and finally launches RTOS tasks to manage individual hardware components. These tasks include battery charging (`BATT`), antenna deployment (`ANT`), and (of note) flash panel activation (`FLASH`), among others. The RTOS schedules these tasks based on the satellite's current state as shown in Figure 3.2.

Ground station Satellite A used two distinct ground station networks to collect downlink telemetry for portions of orbit not reachable by the club's own ground stations. At first, the club solicited amateur radio observations by offering an online upload portal

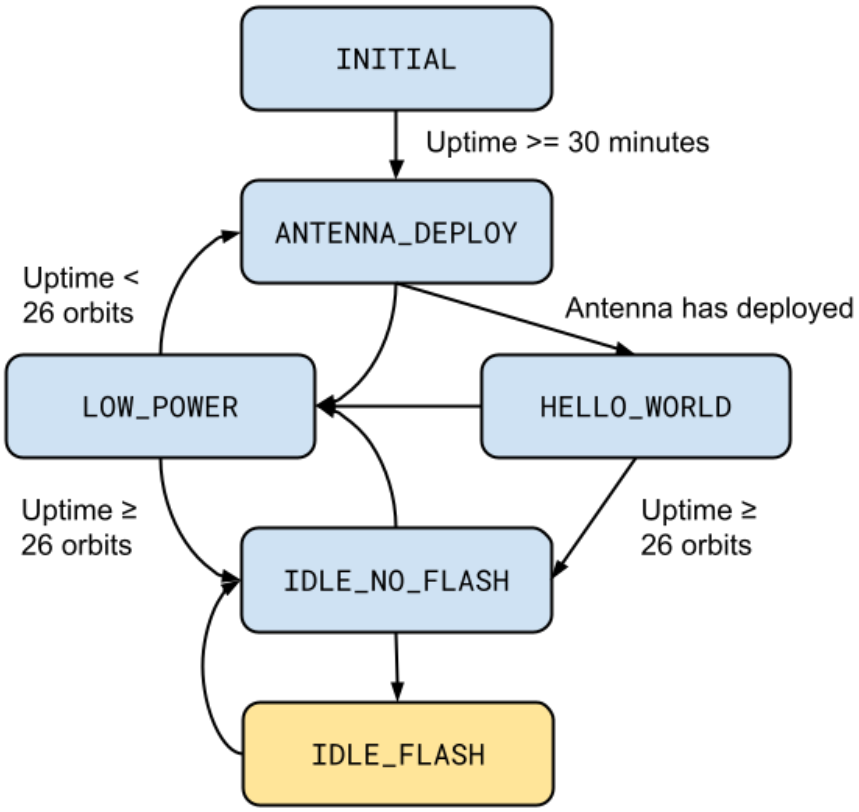


Figure 3.2: Satellite A state transition diagram. The satellite can only flash during the `IDLE_FLASH` state, which is only reachable after a wait period of 26 orbits for safety reasons. State names correspond closely with satellite actions taken during those states, except for `HELLO_WORLD` which acts simply as an idle state. The `LOW_POWER` state is activated any time the satellite detects that the battery levels are too low.

for observations. Later, the club began using a new local decoding feature in the SatNOGS open source ground station network, which required them to have their downlink protocol be publicly accessible. Based on comments in the code and interviews, the club thought of these ham radio operators as honest-but-curious.

Threat Model Comparison

Participant Threat Models As reported in Section 3.5.5, Satellite A designers and NASA had safety concerns related to the flash panel. Satellite A's designers also had *confidentiality* concerns with the satellite's comms protocol, mentioning that they "scrubbed commands" from their public repositories as an "operational security measure." The interviewees discussed various potential adversaries, including an environmental advocacy group who sent them a letter prior to launch and honest-but-curious ham radio operators. Another threat that A2 mentioned was sending commands in such a way that their battery system payload would explode.

Does Participant Threat Model and Code Match? Satellite A's designers successfully mitigated the astronaut-blinding threat by ensuring that the satellite is far enough away from its launch vehicle and the ISS before it can begin flashing. We show the implementation of this mitigation in Figure 3.2. We find evidence in the code that the authors thought about the safety of every state transition. The satellite's "flash immediately" command, for example, does not activate the flash panel unless the satellite is in the `IDLE_FLASH` state.

However, the club's confidentiality expectations were not met. All commands were still available in the code, and all information necessary to build Satellite A's ground station was publicly available during its lifetime. Even if commands were scrubbed, a motivated adversary could reconstruct the protocol based on their unencrypted uplink network traffic. Since the satellite commands are not authenticated, adversaries could have interacted with the satellite, although neither we nor the participants saw evidence that this occurred. We were unable to verify whether sending up a certain set of commands could cause the satellite

to explode.

Does our Threat Model Match? We also believe the most significant risk is physical harm to astronauts, and this outcome is successfully mitigated by precise state transitions. While we do not evaluate the feasibility of the exploding satellite attack, we do think that there are other ways to deny service not discussed by participants. Available commands included simple pinging, rebooting the satellite, enabling/disabling radio responses (presumably to comply with amateur radio laws), enabling/disabling the flash panel, and sending a command to flash the panel immediately. An attacker with a ground station could arbitrarily disable both radio responses and the flash panel, as well as reboot the satellite. If the club does not think to re-enable radio responses or the flash panel they may believe that the satellite has failed in some more severe way, affecting the satellite's *perceived availability*.

Before the satellite decayed, we count several reported CVEs [2] for FreeRTOS 9.0.0. While we did not evaluate whether these CVEs were applicable for Satellite A's configuration, dependence on external code that is not updated regularly is a well-known security risk. Satellite A also lacks a firmware update mechanism, which increases this risk.

3.6.3 Satellite B

Satellite B launched in 2020 and decayed in 2023. This satellite tested a propellant system and a second payload that we omit since it is not relevant to our study and revealing it could facilitate de-anonymization of the club.

Satellite B's security posture heavily relies on *obfuscation*. While the satellite's source code was public during its lifetime, B2 mentioned that extensive protocol layering made it "pretty impossible to send commands." Additionally, the radio implementation (which was provided to the club by amateur radio collaborators) was not public for our analysis. Focusing instead on the satellite's bus architecture, we find that the radio broadcasts packets

[2] `https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=freertos`

directly on a CAN bus for consumption by subsystems. These packets are implicitly trusted by all subsystems, indicating that the radio was the main security boundary for the satellite.

System Overview

Satellite B was constructed out of many individual microcontrollers implementing various subsystems, such as ADCS and propellant control, and connected via a shared CAN bus. The radio system was provided by AMSAT (the Radio **Am**ateur **Sat**ellite Corporation [8]), and was also connected to the satellite's CAN bus; received uplink packets were written directly to CAN and filtered/interpreted by the subsystems, while telemetry packets were downlinked as they were generated by the microcontrollers. The satellite used an open-source high-level operations tool to issue commands that get translated to lower-level messages on the CAN bus.

Threat Model Comparison

Participant Threat Models B1 was in Club B while their satellite launched, while B2 only started after the launch. Both mentioned concern about the thrusters and propellant on-board. B1 did not think that the thruster was strong enough to be used as a projectile. Instead, they were concerned with denial-of-service through the attacker sending up many messages to the satellite, either to flood the comms system or to send commands to use and drain the propellant. B1 thought that the adversary would be a "for-fun hacker."

B2 on the other hand *does* believe that the propellant could have been a problem, and that the worst case was inciting an international incident by either moving the satellite to cause debris or to knock into space vehicles owned by other countries (See Section 4). Because of this threat, B2 thought an adversary might be a foreign nation.

Does Participant Threat Model and Code Match? Satellite B's architecture implicitly places all security expectations onto the radio. The satellite's internal bus is a broadcast CAN bus where every node can see every message, but filters only for messages it can handle. The radio writes all received packets to the CAN bus, presumably after

unwrapping some link-layer framing information. As such, the AMSAT radio would have been our primary focus for code analysis had it been made public. We instead checked AMSAT's documentation for this line of systems and Satellite B's open-source ground station, and did not see evidence of security mechanisms in either. Our participants did not recall there being comms link encryption, further supporting this assumption.

To meet Club B's threat models for propulsion system safety, we believe that command authentication would be necessary to prevent either denial-of-service or an international incident, as envisioned by our interviewees. A2 said that radio sequence numbers could make it harder for adversaries to "patch in a command," but we did not find evidence of sequence numbers being used in either AMSAT's public documentation or Satellite B's ground station. Therefore, there is a mismatch of safety expectations between Satellite B's designers and their code.

Based on interview responses, we believe Satellite B's complex and layered comms protocol led club members to mistakenly believe it was *confidential*; one participant said it was "encrypted by obfuscation" but "not industrial-grade encryption." However, their perceived threats did not necessarily require confidentiality to be successfully mitigated.

Does Our Threat Model Match? We find evidence that the Satellite B repositories were public during its deployment. Since we found no authentication or encryption of commands, we argue that a motivated nearby adversary could have discovered which commands exist and how to send them.

We agree with our interviewees that denial of service and causing an international incident are the worst-case scenarios. We find commands primarily for changing the ADCS magnetorquer settings, powering on/off various subsystems, and pulsing the propulsion system. An adversary on a limited budget (Section 3.6.1) could use these commands to harm availability, e.g. by tumbling the satellite, turning off subcomponents, or exhausting propellant.

3.6.4 Satellite C

Club C implemented a complete open-source amateur satellite system, used so far in two launched satellites. The most recent satellite was launched in 2024.

We find that the main security boundary for Satellite C is the comms subsystem, and that *authentication* (rather than *obscurity*) is the primary security mechanism. We identify two small implementation errors in their authenticity mechanisms and a lack of defense-in-depth. Beyond the comms subsystem, we find advanced functionality enabled by Satellite C's Linux-capable C3. Unique among the satellites we analyzed is a software update capability.

System Overview

Satellite C's architecture consists of a variety of interconnected ARM-based processing units, ranging from microcontrollers running an embedded RTOS to Linux-capable CPUs. These processors are all attached to their relevant subsystems (ranging from ADCS to solar panels) and a central CAN bus for intercomponent communication. The C3 is built as a standard Python application running on Linux, enabling advanced functionality such as software updates.

Satellite C's payloads are a novel high-speed comms link and a camera. As part of the club's outreach goals, they want to take images with the camera from space and then transmit them to mobile ground stations built by local high schools.

Communications Satellite C's comms protocol follows the official CCSDS protocol of the Unified Space Link Protocol (USLP). They use a keyed HMAC built on 256-bit SHA3 for authenticity. Sequence numbers are used to attempt to prevent "repeat attacks," (we believe they meant *replay* attacks) as noted in their public documentation. Their high-speed comms system, operated by the C3, is separate from the main comms link. Commands for taking photos and quickly transmitting them first go through the main C3 computer (following the packet structure), then are immediately routed to and handled by the high-speed subsystem.

Threat Model Comparison

Participant Threat Models C1 identifies both confidentiality and regulatory concerns, stating they did the "bare-minimum of encryption and due diligence". They also expressed concerns related to the camera [65] violating NOAA regulations by taking images of space (and orbiting objects) rather than Earth. Command integrity is required to prevent the risk of an adversary sending a malicious camera command.

Does Participant Threat Model and Code Match? C1's stated threat model is partially addressed in the code. Command integrity concerns are addressed in the comms uplink aside from two implementation issues. We saw no evidence of uplink encryption, contradicting C1. However, based on their concern about violating regulations, uplink confidentiality is unnecessary; only uplink *authenticity* is needed.

We find that Satellite C developers consider the comms subsystem to be the satellite's main security boundary. Each message includes both an HMAC and a sequence number to prevent replay attacks, though there were two slight errors.

First, we found an off-by-one error on packet sequence numbers, allowing for replay of the last received packet an arbitrary number of times until another authentic command increments the internal sequence number. We confirmed that this replay attack works by running the C3 software locally and successfully sending the same packet multiple times.

We argue that sequence numbers only protect commands which change the system state independently of the number of times they are invoked (*idempotent* commands). Our code analysis revealed a rich set of *non*idempotent commands, ranging from component- and system-level resets to real-time clock updates. We did not fully investigate the idempotency of every command (since there are very many), but did check some of the more sensitive ones related to availability. For example, we verified that the magnetorquers are configured via "setpoint" commands that indicate the desired torque level along each axis, rather than specifying *changes* to these levels; this makes the magnetorquer commands idempotent and mitigates against the effects of a potential replay attack.

The other issue is a timing vulnerability on the packet HMAC check due to a non-constant-time comparison operator. An attacker with sufficiently precise timing information could possibly deduce the expected HMAC for an arbitrary packet. We did not check the feasibility of this attack since there are some major limitations, such as the HMAC's 32-byte length and a lack of prior work on ground-to-space timing side channels. Nevertheless, we reserve establishing practical exploitability for future work.

Additionally, we did not find evidence of specific security measures that were taken in the rest of Satellite C's code, indicating that defense-in-depth was not a concern for the developers (which also matches C1's threat model). For example, we found that no integrity or authenticity checking is done on update files beyond the HMAC/sequence number checks as update packets are received. Therefore, if an attacker were able to bypass these checks they may be able to establish persistence on Satellite C via a malicious software update. Similarly, we find no specific security measures taken to prevent illicit imaging of other satellites. Instead, the camera is simply triggered by a command from the C3 when a corresponding radio command is authenticated.

Does our Threat Model Match? We agree that regulatory concerns in space are important to consider. Like other satellites, another potential threat is denial of service. The replay attack vulnerability could harm satellite availability.

3.7 Security Recommendations for Smallsats

Informed by our interviews, code analysis, and our own satellite club experience, we discuss approaches that could help improve security practices among university and smallsat teams. Some recommendations apply to other amateur satellites and even industry, as our participants and prior work [163] show industry security also needs enhancement.

3.7.1 Open-Source Tools and Techniques

Software/Hardware Any tool should consider the incentives impacting the satellite development process, including the particular emphasis on satellite systems' reliability (Barrier 9). Any proposed smallsat security tool would ideally have reliability benefits. An example of an effective security and reliability tool is updating the hardware of a commonly used smallsat component. We found that multiple university clubs rely on the open-source OpenLST transceiver package[3] for their communications. One comment in OpenLST code is "TO-DO: handle encryption". In their documentation they say that they do not implement encryption. A future project could be to fork this repo and update it with encryption and authentication. Since OpenLST is on longer maintained, hardware updates alongside security improvements would increase reliability and thus likelihood of adoption.

As university clubs prioritize learning, a fully-furnished secure smallsat framework may have limited appeal. However, an out-of-the-box cryptographic comms solution could be valuable, as comms systems often define security boundaries. Authentication-only tools may also be useful as confidentiality is not applicable to every club's threat model.

Checklists Security considerations in smallsat repositories are incomplete. B2 suggested a security checklist. This could include recommendations for cryptographic libraries, secure reference code, design guides, and organizational security practices. A2 raised concerns about cryptographic techniques affecting reliability, so the checklist could integrate lessons from embedded software and network security.

3.7.2 Operational

Actions Within Club We saw partial threat modeling in Club A and Club C. Satellite clubs could incorporate more thorough threat modeling into the design stage. Clubs could use tools such as threat modeling cards [38, 115]. Though suggesting tools does not diminish the time constraints clubs face.

[3] `https://github.com/OpenLST`

Third-party Actions To address these competing incentives, a trusted third party, such as NASA, could recommend or require security measures before launch. NASA's Cubesat Launch Initiative Program [117] would be a particularly good location to promote good security practices in educational and non-profit satellites, since they are involved in the process of many universities' smallsat development. NASA could provide some information about the importance of smallsat security in their educational resources, such as their Cubesat 101 book [26], or even incorporate the secure design checklist suggested above. For those not involved in NASA CSLI, launch providers could also set some requirements for satellites. Trusted third parties could also offer threat modeling support or serve as security contacts for university teams.

A "No Encryption, No Fly Rule" was proposed by authors based on their analysis of the damage propulsive systems on smallsats could cause [88]. We propose expanding this rule to cover integrity and authentication, since these are even more important than confidentiality for many amateur smallsats.

3.8 Summary

In this chapter, we examined the security practices and perspectives of university smallsat teams through interviews and code analysis. While teams implemented some security measures, they were neither systematic nor complete. We observed impressive engineering efforts and believe improvements in tools, organizations, and third parties involved in launch can improve smallsat security. These changes will hopefully align with teams' goals of reliability and education while mitigating risks to the broader space community.

Chapter 4

Security Impacts and Defenses of Compromised Satellite Peripherals

The vast majority of satellites in the past used to be highly bespoke systems. Now, more and more satellites are manufactured as parts of fleets (such as with satellite internet) or via companies that sell buses at scale to customers. Thus, more hardware and software is being shared across satellites than ever before. This increases efficiency, but also makes these complex satellites an even more enticing target for adversaries if one attack were to be able to take down multiple satellites at once. Now, high-resource adversaries like states might be more likely to invest resources into a supply chain attack to ensure that a compromise gets on a group of target satellites. For example, a state adversary could sell compromised peripherals to a customer looking for affordable components. Spurred by this trend of satellite software and hardware at scale, Chapter 4 addresses two main questions:

1. Can today's satellites effectively respond to a malicious peripheral attack?
2. If today's satellites cannot handle such an attack, is it possible to develop a solution that can?

Part I of this chapter aims to answer this first question. This chapter uses a malicious star tracker to show that existing telemetry anomaly detection systems, like the NASA cFS limit checker, are not built to handle malicious cases. This part investigates attacks along two axes—maliciously incorrect timing of data and maliciously incorrect content of data—and shows that these attacks can cause mission failure and lead the satellite to tumble, or rotate uncontrollably. The work from Part I initially appeared at NDSS SpaceSec in 2026

as the paper "The Compromised Satellite Peripheral Dilemma" [113].

Part II addresses the second question. We propose ControlPhoenix, a novel sensor diagnosis approach aimed at identifying and responding to malicious peripheral attacks on satellites and other cyber-physical systems. We instantiate ControlPhoenix in an attitude determination and control loop on an extensive, high-fidelity satellite simulation, and show that it correctly handles multiple types of stealthy attacks with minimal space and developer overhead. The work from Part II is in collaboration with Tadayoshi Kohno, Samuel Mergendahl, and Richard Skowyra.

Chapter 4 Part I: The Compromised Satellite Peripheral Dilemma

4.1 Motivation

An unprecedented number of commercial satellite launches [155] has recently led to the emergence of space systems as an unofficial category of critical infrastructure. These commercial systems provide high-speed, low-latency communications [104], position, navigation, and timing (PNT) [70], weather forecasting [108], military defense [153], and geospatial imagery [90], among other applications. Commercial satellites deploy most frequently into Low Earth Orbit (LEO) due to lower launch costs [68] as well as the desire to use commercial-off-the-shelf (COTS) components, which are less suited to the longer mission lifespans and harsher conditions of higher orbits [22].

Given the critical nature of space systems, cybersecurity is of utmost importance. Satellite security has traditionally focused on the protection of communication links [150], but more expansive threat models have been proposed recently [32, 75]. For example, the threat of memory corruption should concern satellite designers since attackers can leverage these common vulnerabilities to launch advanced code reuse attacks and circumvent many defenses.

One understudied aspect of satellite system security is the inherent trust of the microcontroller software found in satellite peripheral subsystems. As space companies increasingly

rely on outsourcing components to decrease cost and development time, the risk of software supply-chain attacks originating from peripheral subsystems also increases. Common satellite peripherals include star trackers, reaction wheels, or other sensors and actuators, and the software running in these peripheral subsystems is often proprietary and treated as a black-box by mission designers. This can lead to situations where the flight computer places undue trust in the code and data of the peripheral subsystems. While some satellite security work has touched on the risk of software supply chain attacks on satellites [75], as well as the risk of external physical disruption of satellite sensors [32], the impact of a compromised peripheral subsystem on the flight computer has not been experimentally evaluated. At the same time, adversary incentives for such attacks have increased due to emergence satellite fleet and bus companies that produce at-scale. A supply chain attack on satellites could now impact an entire fleet at once.

In this chapter, we explore the impact of a compromised peripheral subsystem on a satellite. We investigate a typical Guidance, Navigation, and Control (GNC) subsystem on a spacecraft that uses a star tracker sensor peripheral, an Attitude Determination and Control (ADCS) feedback loop, and reaction wheels as actuator peripherals. We demonstrate that without any protections, a malicious star tracker peripheral can cause GNC failure through easily executed attacks. Specifically, we demonstrate that the ADCS software component on the on-board computer acts erroneously when the sensor peripheral either sends *corrupted data* or nominal data at *unexpected frequencies*.

Additionally, we show that two commonly deployed software resiliency strategies are insufficient against a malicious peripheral. Namely, we study the detection of anomalous sensor input against configurable thresholds, and secondly, the use of sensor fusion with a redundant set of sensor peripherals. Historically, these strategies are believed to provide resilience against faulty components. However, we find that neither strategy is effective against *malicious* components.

We observe a design trade-off for system configuration on satellites given these resiliency strategies: Either the system deploys countermeasures against malicious peripherals and

pays a cost in nominal mission performance, or the system cannot survive against malicious peripherals. This dilemma suggests that satellite operators face significant challenges if they use existing tools to defend against malicious peripherals.

Previous work has mentioned the dilemma of attack detection sensitivity and its impacts on reliability and runtime [167], but this problem has yet to be considered in the satellite context. Lu et al. highlights the need for more domain-specific work on cyber-physical system attacks and recovery research, as each domain has its own constraints and challenges [101], and we aim to address this gap by providing evaluation of possible attack scenarios in the satellite domain. The satellite domain poses different constraints compared to other robotic vehicles like drones, a common platform for testing sensor attacks and associated recovery techniques [34, 35, 98]. Compared to drones, satellites must handle a much longer mission, taking up to multiple years. Satellites must be able to handle very stealthy attacks that may only cause measurable impacts over periods longer than a single drone mission. A satellite's location must be precise to stay in the correct orbit while pointing its communications module and sensors in the correct direction, whereas a drone might be able to land many meters away from its target landing destination while still executing a safe mission. Other unique considerations for satellites are their possible vulnerability to hardware faults due to radiation in orbit, and the complexity of multiple control systems and sensors operating on a machine with limited power and computation to put towards attack detection.

After we provide the relevant background necessary in Section 4.2 and introduce our studied threat model in Section 4.3, our work makes several contributions:

- We introduce a series of malicious behaviors that a compromised peripheral can employ as an attempt to degrade a satellite mission in Section 4.4. We show these for the first time on a satellite system within the emulated system described in Section 4.5.
- We characterize the impact of a malicious satellite peripheral micro-controller on the on-board flight computer in Section 4.6.
- Further, in Section 4.6, we implement and evaluate two resiliency strategies commonly

used in satellite missions and highlight their limitations.

- We argue that existing anomaly detection and recovery systems are insufficient for the satellite context in our attack model.

4.2 Background

Satellite System Architecture. Satellite systems typically consist of a central flight computer that interfaces with several peripherals. The flight computer orchestrates communication between each of the peripheral subsystems, hosts software to perform *soft* real-time control loops, and performs command and data handling tasks. Peripherals may have responsibilities ranging from non-volatile storage to mission-specific sensors and actuators. As separate hardware components connected to the on-board computer over a serial link, these peripherals contain their own microprocessors which manage the low-level, *hard* real-time control loops specific to that peripheral. Many peripherals on a satellite interface to sensors and actuators to enable the flight software on the central flight computer to interact with the physical environment.

Guidance, Navigation, and Control. Star trackers are sensors which are employed on satellite buses to gather information about the spacecraft's attitude (orientation). A star tracker is an optical sensor which is purpose-built for capturing starfield images and computing attitude solutions using a database of known celestial bodies. The result of a star tracker measurement is an attitude quaternion describing the orientation of the spacecraft. This is used to inform an attitude estimate as part of a sensor fusion strategy, which combines data from one or more star trackers and other sensors to reject noise and achieve higher precision. Moreover, reaction wheels are often deployed to adjust the satellite orientation as needed. See Figure 4.1 for a high-level view of how a star tracker might work, with potentially malicious actors in the sensor data pipeline.

NASA core Flight System. The core Flight System (cFS) [41] is a modular flight software framework developed by NASA. cFS is written in C and uses a message passing architecture over a software bus to facilitate communication between applications. Each application in cFS is separately compiled and runs in a dedicated thread. The modular design of cFS contrasts sharply with traditionally designed flight software systems, which typically have monolithic architectures. The modular nature of cFS benefits mission reusability.

The limit checker is an optional cFS application that monitors software bus telemetry for anomalous values according to a configurable set of *watchpoints* and *action points* [42]. The watchpoints define which application messages (and their associated values) to monitor. The action points define what conditions indicate anomalous values and specify the relevant remedial actions to take under any met conditions, such as ignoring future packets from the anomalous source.

Satellite Security Research. There exist many challenges to implementing security in the satellite context, including the inability to physically access the satellite post-launch, emphasis on safety and availability over confidentiality, and handling radiation faults [75].

Recent work has shown that satellites in industry and academia practice insufficient security [111, 163]. Jero et al. propose future approaches to a more in-depth security posture beyond communication security, such as reverting to a "safe mode" [75]. However, safe mode typically does not prevent malicious peripheral attacks, as it may inherently trust important sensors like the star tracker.

While previous satellite security research has touched on supply chain security [44,75,126], to our knowledge none have evaluated the details of what such a supply chain attack entails, including whether firmware or software is compromised, how and when the satellite might be compromised, or the persistence of such a compromise. For example, Pavur et al. list a "threat matrix" toolbox through synthesis of previous satellite security research [126]. This toolbox does not include any mention of malicious peripherals or the impact of incorrect sensor data through stealthy attacks. However, Pavur et al. does highlight institutional

barriers leading to the fact that "almost no technical research exists on the defense and monitoring of systems in orbit", and that this is likely "due to substantial logistical barriers involved in accessing and conducting research with representative space hardware" [126]. We overcome this barrier through extensive effort in building out a representative satellite simulation using NASA cFS along with configurations for different peripheral attacks and detection methods.

4.3 Threat Model

We assume a typical satellite system architecture in which an on-board computer deploys flight software such as cFS in order to manage multiple peripherals connected to the on-board computer over a serial link. Specifically, there exists three COTS star trackers and multiple reaction wheel hardware peripherals each with their own micro-controllers connected to the on-board computer that operates the Attitude Determination and Control (ADCS) soft real-time loop for Guidance, Navigation, and Control (GNC). To improve the reliability of the system, the ADCS component performs sensor fusion on the redundant star tracker data, and a limit checker application monitors the star tracker data for signs of anomalies. Additionally, we assume one star tracker to be compromised. This could be due to a malicious supply chain attack given the increasing reliance on blackbox COTS components, or some other external force such as sensor spoofing [32]. From the compromised star tracker peripheral, the adversary attempts to degrade the overall attitude of the satellite. See Figure 4.1 for more details.

In a compromised supply chain scenario—which is the main threat scenario for this project—the peripheral has access to the true sensor measurements. Because of its malicious firmware onboard, the peripheral can construct its measurements based on knowledge of the true sensor values. This enables a finer-grained attack that differs from an honest but faulty setting, where the sensor values or timing at which values are sent are not intentionally constructed.

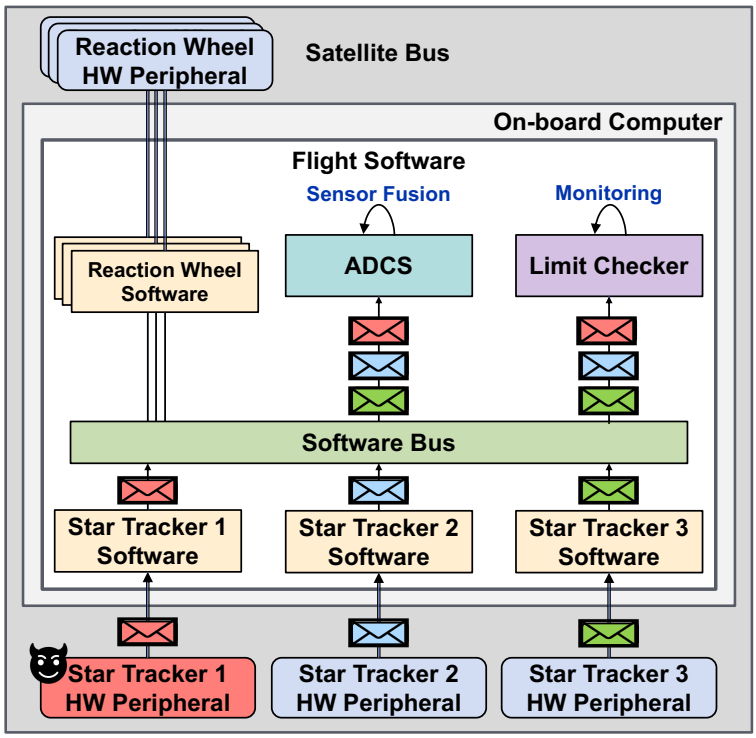


Figure 4.1: Threat model of malicious peripheral.

4.4 Attacks and Common Mitigations Overview

In this section, we introduce two types of attacks that a compromised satellite peripheral could launch, as well as two resiliency strategies commonly deployed in space system software. As mentioned in Section 4.3, we contextualize these attacks and defenses with respect to the GNC subsystem of a satellite.

Attack Impact. In a real satellite mission, incorrect attitude data can cause the satellite to be incorrectly positioned, thereby moving the payload and communications module away from where they should be relative to Earth. This can ruin the mission by preventing collection of the correct data (e.g., if the payload is a sensor collecting images), preventing the satellite from pointing its solar panels properly to the sun and thus losing power, or causing the satellite to lose communication with its ground station due to incorrect radio positioning.

Incorrect sensor data can not only cause the satellite to point in the wrong direction, but can also cause the satellite to tumble (i.e., spin uncontrollably).

Studied Attacks. We separate our attacks into two axes: attacks related to manipulating time without any malicious data manipulations, and attacks sending incorrect data without any malicious time manipulations. The first attack we evaluate is a **Manipulated Data Attack**, where the compromised star tracker peripheral sends incorrect data values. The second attack we evaluate is a **Manipulated Time Attack**, where the star tracker data is correct, but sent at an unexpected frequency. Specifically, in each time period, we withhold the N star tracker data points and instead send them as close together as possible in a burst of size N. Of course, an adversary could combine these two attack techniques (e.g., flooding maliciously crafted data), but we study the disjoint version of the attacks in order to ensure one technique does not influence the impact of the other technique.

Satellite Reliability Techniques. When a satellite developer is preparing for a mission, they often value reliability over security [111]. In particular, the nominal behavior of a satellite must work extremely reliably, but developers must also prepare for issues such as radiation-induced faults. To survive faults and increase overall reliability, satellite developers often use multiple approaches. The first common reliability approach on satellites is to use redundancy. For example, a GNC subsystem might deploy three different sensors each separately determining orientation, and fuse together measurements from all three sensors to output a more robust estimate of the true orientation of the satellite. Indeed, in our evaluation (Section 4.5), we simulate data from three star trackers, where only one is malicious. While redundancy is useful for recovering from faults in the nominal case, using redundancy when at least one sensor is malicious can be ineffective, as shown later in Section 4.6.

The second reliability approach commonly used is a threshold-based limit checking scheme to monitor sensor data for analogous conditions. For example, the NASA cFS flight software contains the Limit Checker application for this purpose. The limit checker is used to detect

when values are out of range and trigger action on those detections. For example, the limit checker can have a watchpoint that detects when the temperature of a satellite's battery is too high, and can trigger an action to cool the satellite. Or, a watchpoint could verify that the satellite's star tracker data is within the expected range for its unit of measurement (between -1 and 1 for quaternions), and shutdown the sensor if that range is not met, with the assumption that a temporary fault may lead to an out-of-range value. While useful in honest settings, we show in our results that a limit checker is poorly suited for handling malicious peripherals.

4.5 Experimental Setup

In this section, we describe our satellite emulation environment and test flow for launching the attacks introduced in Section 4.4.

4.5.1 Emulation Framework

We emulate a typical GNC subsystem using a software-only test harness that deploys cFS on Linux as the flight software for the on-board computer. The spacecraft is instantiated within the Basilisk astrodynamics framework [78], actively used in research and industry for testing, which simulates the rigid body dynamics of the vehicle in Low Earth Orbit (LEO). Additionally, cFS deploys custom applications from Basilisk that implement a simple attitude control loop using emulated sensors and actuators to create a closed-loop emulation of the satellite.

In particular, a star tracker application ingests attitude state information from the simulation to model the behavior of a physical star tracker sensor. The sensor data is forwarded to the ADCS application, which uses an attitude control algorithm from Basilisk to compute the reaction wheel torques necessary to maintain an attitude reference. A reaction wheel application consumes the torques and forwards them to the simulation, where they are integrated into the rigid body dynamics to emulate the behavior of physical

reaction wheels. The simulation feedback loop through cFS executes at a rate of 1 Hz and achieves stabilization to a given reference attitude within 30 seconds under honest (nominal) conditions. This simple attitude control test harness allows for the execution of maneuvers by sending a command to change the attitude reference.

Because we are more interested in the impact of a compromised peripheral—rather than how it became compromised—the test harness also includes commands to trigger attacks from the malicious star tracker, as well as system configurations to adjust the settings for the deployed resiliency mitigations.

4.5.2 Attack Setup

The procedure is as follows for both manipulated time and manipulated data attacks:

1. *Step 1: Ensuring limit checker metrics meet honest sensor requirements.* Before performing any attacks, we first define the limit checker configuration based on the distribution of expected nominal data and the level of acceptable risk for the mission (e.g., tighter min and max limit thresholds are more conservative against faults, but may cause degradation of nominal performance). This procedure represents the actions a satellite developer might take to improve the satellite's reliability. We only tune the limit checker for star tracker data so as to only focus on one satellite subsystem.
2. *Step 2: Perform attacks.* We then perform attacks that 1) operate within these limit checker thresholds so as not to trigger any action, and 2) operate slightly out of those thresholds to show the impact of detecting an attack that is not stealthy. We show that operating within these thresholds can still lead to mission failure, and even detection of non-stealthy attacks has performance slowdowns.

We evaluate these limit checker settings and attacks both with and without sensor redundancy, and show that redundancy cannot necessarily help survive attacks, even if only one out of three star tracker sensors are malicious.

4.5.3 Testing

The test flow is as follows. First, a maneuver is triggered by an attitude change command to reach a target attitude after starting from a different initial attitude. Then, after a time delay, a command triggers the attack. If the satellite has not achieved the reference attitude to a given tolerance after 120 seconds, the simulation is terminated. In particular, our tests attempt an attitude maneuver from starting quaternion values of

$$q_0 = [1, 0, 0, 0]^T$$

to the target quaternion values of

$$q_1 = [0.6002842, 0.4617571, -0.4617571, -0.4617571]^T$$

This target orientation was chosen without loss of generality and at random.

4.6 Attack Results

In this section, we leverage the emulation framework from Section 4.5 to report the impact of the data manipulation and timing attacks. Additionally, we report how typical resiliency defenses will affect honest (nominal) and malicious (attack) mission scenarios. Results are averaged over 5 trials for all tables, where each entry of the table is the area under the curve of the attitude error metric described below. A smaller value in a table entry means a more successful defense as it represents less cumulative attitude error. The reader can refer to the appendix for example-based visualizations of attack impacts.

4.6.1 Attitude Error Metric

Attitude error is defined as the angular displacement in radians of the satellite's $SO(3)$ orientation relative to the goal orientation, q_1. Error is computed by finding the magnitude of the axis-angle representation [161] of the quaternion quotient between current and goal orientation, as follows:

$$\Delta q(t) = q(t)q_1^{-1}$$
$$\Theta(t) = 2\text{acos}(\Delta q_w(t))$$

Where:

$$q(t) : \text{Quaternion attitude time series}$$
$$q_1 : \text{Goal orientation}$$
$$\Delta q(t) : \text{Orientation relative to goal}$$
$$\Theta(t) : \text{Angular displacement (radians)}$$

The attitude error is taken as $\Theta(t)$, the angular displacement relative to goal. For coarse-grained evaluation of three-axis stabilization performance under various attack scenarios, an additional area-under-the-curve (AUC) metric is defined as $\Theta(t)$ integrated through time:

$$\text{AUC} = \int_{t_0}^{T} \Theta(t)dt$$

Where t_0 is the fixed time offset into the maneuver at which the cyber attack is launched. T is a predefined timeout value at which the satellite is known to achieve its goal orientation under nominal conditions, with no cyber attack taking place. AUC is computed using rectangular Euler integration for each simulated scenario. It is assumed that AUC takes on some ideal, minimal value under nominal conditions when the satellite converges smoothly to the goal orientation. Given that $\Theta(t)$ is non-negative by definition, any sub-optimal deviation from the expected trajectory manifests as an increase to AUC. Therefore, AUC may be taken as a metric for evaluating the degree of disruption to the three-axis stabilization in the presence of a cyber attack. We use the AUC measurement in Tables 4.1 through 4.10.

Table 4.1: Honest case: turn star tracker off if quaternion difference is *smaller* than δ_{min} for n consecutive times. NO redundancy defenses

	Consecutive limit checker alarms before response				
δ_{min}	2	4	6	8	10
0.00001	5.127	5.127	5.126	5.127	5.127
0.0001	5.126	5.127	5.127	5.127	5.127
0.001	5.127	5.127	5.127	5.127	5.127
0.01	118.369	5.127	5.127	5.127	5.126
0.1	168.598	118.369	5.127	5.127	5.127

Table 4.2: Honest case: turn star tracker off if quaternion difference is *smaller* than δ_{min} for n consecutive times. WITH redundancy defenses

	Consecutive limit checker alarms before response				
δ_{min}	2	4	6	8	10
0.00001	5.133	5.132	5.132	5.132	5.133
0.0001	5.133	5.133	5.132	5.132	5.132
0.001	5.132	5.133	5.133	5.133	5.132
0.01	117.794	5.133	5.132	5.132	5.133
0.1	168.677	117.801	5.133	5.132	5.133

Table 4.3: Honest case: turn star tracker off if quaternion difference is *bigger* than δ_{max} for n consecutive times. NO redundancy defenses

	Consecutive limit checker alarms before response				
δ_{max}	2	4	6	8	10
0.00001	168.598	118.368	5.127	5.127	5.127
0.0001	168.599	118.370	5.127	5.127	5.127
0.001	168.595	118.368	5.127	5.127	5.127
0.01	168.592	5.126	5.127	5.127	5.127
0.1	5.127	5.127	5.126	5.126	5.127

Table 4.4: Honest case: turn star tracker off if quaternion difference is *bigger* than δ_{max} for n consecutive times. WITH redundancy defenses

	Consecutive limit checker alarms before response				
δ_{max}	2	4	6	8	10
0.00001	168.669	117.801	5.133	5.133	5.133
0.0001	168.672	117.809	5.133	5.133	5.133
0.001	168.664	117.806	5.133	5.133	5.132
0.01	168.677	5.133	5.133	5.133	5.133
0.1	5.132	5.132	5.132	5.132	5.132

Table 4.5: Data attack: turn star tracker off if quaternion difference is *bigger* than δ_{max} in columns or *smaller* than δ_{min} in rows. Consecutive failures=6. NO redundancy defenses. N/A if $\delta_{min} \geq \delta_{max}$

	δ_{max}			
δ_{min}	0.0001	0.001	0.01	0.1
0.00001	161.188	162.500	191.670	228.594
0.0001	N/A	162.813	192.109	228.641
0.001	N/A	N/A	194.488	226.909
0.01	N/A	N/A	N/A	229.570

Table 4.6: Data attack: turn star tracker off if quaternion difference is *bigger* than δ_{max} in columns or *smaller* than δ_{min} in rows. Consecutive failures=6. WITH redundancy defenses. N/A if $\delta_{min} \geq \delta_{max}$

	δ_{max}			
δ_{min}	0.0001	0.001	0.01	0.1
0.00001	156.508	151.699	74.316	80.627
0.0001	N/A	151.111	74.528	80.626
0.001	N/A	N/A	107.254	79.744
0.01	N/A	N/A	N/A	109.540

4.6.2 Data Manipulation Attack Results

We evaluate the impact of malicious sensor values on the attitude of the satellite. We study two defense strategies against this attack: first, redundancy of three star trackers (where only one is malicious), and second a defense of detecting when the variation between consecutive quaternion values is too high or low. In this section, we primarily study how an operator sets and applies limit checks to flag any unsafe *variation* of quaternion values. Each test also applies a limit check to ensure the raw quaternion values are sensible (i.e., any quaternion values outside the range of $[-1, 1]$ are flagged). These are reasonable thresholds to set on a mission, as nonsensical attitude values or dramatic attitude shifts both could damage the satellite and represent faulty behavior.

Determining Thresholds for Honest Case

We first run functional tests on different possible limit checker configurations and redundancy scenarios when the star tracker(s) are honest. Table 4.1 shows the attitude error after running the functional test without any redundancy and only one single honest star tracker. The table rows correspond to the limit checker settings of triggering an alarm if the data values changed by *less than* the threshold δ_{min}. The columns correspond with how many times in a row this threshold δ_{min} is seen before the limit checker marks the star tracker as malicious and turns it off. Most configurations in the table didn't impact nominal operations, but assuming δ_{min} of less than 0.1 or less than 0.01 led to mission failures when the consecutive failure count was too low. Thus, it would make sense for the mission operator to set the δ_{min} values to something like 0.001 or smaller. This is the most sensitive these thresholds could possibly be; in a real mission, the satellite developer would likely have to relax these thresholds even further to handle all nominal operations, making possible attacks that work within these scenarios easier to execute in a real mission setting. The results in Table 4.2 are similar given that the data is the same across star trackers, aside from noise applied to the quaternions that was sampled from a normal distribution with a standard deviation of

0.1.

On the other end, Table 4.3 and Table 4.4 correspond to the limit checker settings of triggering an alarm if the data values changed by *greater than* the threshold δ_{max}, without and with redundancy respectively. The safest bet for detecting faults while still handling the nominal setting would be to set $\sigma_{max} >= 0.1$, and set the consecutive failure configuration to ≥ 6, though higher would likely be better to avoid being too sensitive for other nominal attitude maneuvers. As in the case with δ_{min} configurations, redundancy does not make a noticeable difference in the response to different thresholds since the data across star trackers is close to the same given the amount of data noise in the simulation.

Data Manipulation Attacks within Honest Thresholds

Next, we perform a manipulated data attack that stays within the allowable honest limit checker settings derived from the previous step. Given the results in Table 4.1 through Table 4.4, we show the limit checker consecutive failures setting of 6 for Table 4.5 and Table 4.6. However, we note that regardless of how many consecutive failures the operator allows, the attack is able to evade detection completely. To perform a stealthy attack, the malicious star tracker increments or decrements each quaternion value by the average threshold between the big and small threshold: $\delta_{attack} = \frac{(\delta_{min}+\delta_{max})}{2}$. It continues incrementing by δ_{attack} until it reaches the limit of quaternion value 1, then begins decrementing by δ_{attack} until it reaches -1. This attack leads to failure of every functional test, i.e., the satellite is never able to achieve its target orientation. Tables 4.5,4.6 show the effectiveness of this attack. Despite tight constraints tuned for the honest setting, the stealthy attack achieves full mission compromise.

Limit Checker and Redundancy are Insufficient for Manipulated Data Attack

In the manipulated data attack, the limit checker flags a star tracker as anomalous only if its reported values are outside of normal (e.g., not between -1 and 1), or if the *change* in quaternion values are greater than a certain threshold between time t and $t+1$ or less than a

certain threshold. Of course, this particular installation of the manipulated data attack could be defended against if we also implement a defense that checks that the pairwise difference between star trackers is within a certain range [125].

However, we argue that continuously adding more static limit checker policies in a "cat-and-mouse" game with the attack is generally ineffective. Whatever thresholds do get set in the limit checker, a malicious star tracker could just send data that is above the minimal threshold and below the maximal threshold. This motivates the need of a new anomaly detection technique for satellite operators that extends the capabilities of today's limit checker applications. For the scenario of pairwise differences, this would only work if only one of the three star trackers is compromised. Additionally, redundancy techniques often have a variety of sensor types, such as determining satellite orientation through fusing data from a star tracker, sun sensor, and earth tracker. And Park et al. note that the pairwise difference defense they propose may not hold up in stealthy attack scenarios [125].

Additionally, the limit checker has a maximum table size, as it must react to every single message and determine actions for every single watchpoint. The limit checker default size is not large enough to write multiple threshold values for every single sensor value and possible anomalous scenario. Even if the limit checker were large enough, writing possible thresholds for many different attack scenarios would put far too much burden on the developer for it to be a practical defense.

4.6.3 Time Manipulation Attack

Next, we investigate the Time Manipulation Attack where the compromised satellite peripheral withholds and then bursts (non-manipulated) data. In particular, we show that bursty communication from a malicious peripheral causes either 1) failure to successfully adjust the attitude of the satellite or 2) degradation of attitude maneuver performance on the satellite. We also elaborate further in this section that the limit checker works poorly for time frequency thresholds.

Determining Timing Thresholds for Honest Case

Timing inconsistencies under no malicious adversary could be caused by different factors. One issue more likely in space than on Earth are radiation-induced faults [40]. Another issue is hardware aging [9]. A satellite developer may try detecting faults by monitoring the frequency at which data is sent. We thus test the limit checker at various levels of frequency at which to take action on the star tracker on differing levels of timing noise in the honest setting.

Tables 4.7 and 4.8 show results of honest sensors at different noise levels and different limit checker thresholds. Timing noise in actual star tracker peripheral hardware is difficult to quantify, especially due to the variety of star tracker options at different price points, changes in timing behavior as hardware ages during a multi-year mission, and the difficulty of determining the exact impact of radiation on different hardware. Therefore, we test on a subset of possible honest noise values.

The columns of Tables 4.7 and 4.8 correspond with how much timing noise the honest star tracker exhibits. The standard deviation is a half-normal distribution starting at 1 and truncated at 6. In other words, the burst amount can be between 1 (normal 1hz data timing) and 6, where the data would be sent in a burst of 6 data points in a row after 6 seconds.

Based on experimentation, 0.1 seemed like the most reasonable standard deviation to set an honest, non-faulting star tracker to. This noise amount was low enough that a burst of two data points or higher in the honest noise setting did not occur during testing, though it might experience multiple bursts over a multi-year satellite mission. The following timing attacks thus considered scenarios where the honest star trackers were set to a noise standard deviation level of 0.1. While timing jitter is less likely under low noise conditions, a developer would still want to tolerate small levels of noise so as not to trigger a star tracker shut-off/restart if the timing jitter is only temporary. Therefore, in the honest setting, a developer might not want to trigger an automated limit checker response after a burst of two data points at once, but may want to respond if there is a burst of three. We next perform manipulated time

Table 4.7: Honest case: turn star tracker off if there is a burst of n or more data points, where row is n. Column is standard deviation of burst size . NO redundancy.

Burst	Timing noise standard deviation		
threshold	0.1	0.25	0.5
2	5.127	4.828	172.859
3	5.127	4.828	5.189
4	5.127	4.828	5.189

attacks and show that this burst of three limit threshold can lead to unacceptable mission slowdown, despite the attacker only being able to stealthily send bursts of two at a time.

Time Manipulation Attacks under Honest Thresholds

The manipulated data attack sends a burst of n data elements after n seconds, and no other data from the malicious star tracker in between those n seconds. Table 4.9 and Table 4.10 show the effect of bursty attacks at different burst sizes when there is a single peripheral or multiple, respectively. The burst size in each column represents at what burst frequency a limit checker flags the star tracker. The rows correspond with the actual burst size of the attack occurring.

Even for attack burst size 2, performance degrades in both Table 4.9 and Table 4.10. Even when there is a majority of honest star trackers as in Table 4.10, a burst size of higher than 3 allows the single compromised peripheral to degrade the satellite. The runtime for the functional test even with two other honest star trackers is increased by $\approx 3.4\%$. Due to realtime constraints of the satellite, even a 3.4% increase in runtime can sometimes be catastrophic for a mission.

Table 4.8: Honest case: turn star tracker off if there is a burst of n or more data points, where row is n. Column is likelihood of a burst (standard deviation). WITH redundancy.

Burst	Timing noise standard deviation		
threshold	0.1	0.25	0.5
2	5.133	4.971	5.125
3	5.132	4.933	4.970
4	5.132	4.933	4.970

Table 4.9: Timing attack: turn star tracker off if there is a burst of n or more data points, where row is n. Column is what frequency triggers star tracker shutoff. NO redundancy. For example, the first row shows results for error area-under-curve of a bursty attack of size 2, where the burst frequency threshold shuts the star tracker off if it bursts in size 2 (leftmost column) to size 4 (rightmost column).

Burst	Burst limit checker threshold		
size	2	3	4
2	191.979	5.645	5.645
3	175.372	175.376	38.605
4	179.370	179.383	179.382

Table 4.10: Timing attack: turn star tracker off if there is a burst of n or more data points, where row is n. Column is what frequency triggers star tracker shutoff. WITH redundancy.

Burst	Burst limit checker threshold		
size	2	3	4
2	5.176	5.170	5.170
3	5.193	5.192	5.199
4	5.232	5.235	5.236

Limit Checker and Redundancy are Insufficient for Manipulated Time Attack

Neither redundancy nor the limit checker are adequate to fully prevent the impacts of manipulated time attacks, while also handling temporary timing jitter in the honest setting, leading to a dilemma for the developer of whether to prioritize reliability or security.

Perhaps this could be fixed by having the limit checker check consecutive sets of bursts. For example, let us assume the operator thinks more than two messages per second should lead to immediate action. The operator could add custom code to check this, as we did in our limit checker tuning. They may deem one or two bursts in a row as honest and no action is needed, but they may want to take action if there are more bursts in a row. Unfortunately, consecutive failures for timing attacks will not be detectable due to the way the cFS limit checker is architected. Given that the limit checker must evaluate every star tracker message if it is subscribed, it must return a pass or fail for each data point when a watchpoint is encountered. This means that if $n-1$ data points are seen in a second, the limit checker counts this as a pass. When n data points in a burst are seen, this is one failure. But then when the next data point is seen, this is n seconds after the previous burst, which means that the limit checker would return a pass, and therefore the consecutive failure count returns back to 0. Therefore, the satellite developer can only trigger action upon a single burst, or

trigger no action at all.

4.6.4 More Limit Checker Security Challenges

The NASA cFS limit checker is set to listen to the message id's that a developer specifies. It watches for these message id's, and compares these to threshold values. As the table of watchpoints scales up, simply having watchpoints and possible responses may lead to performance impacts, especially for smaller satellites. In the case of our limit checker thresholds, we had to write custom comparison functions for both checking the difference between data values and the timing frequency of values. By default, the limit checker is only designed to handle very simple checks, such as checking that the star tracker quaternion values are in range. Writing these custom functions, not to mention thinking through all possible peripheral attacks, becomes burdensome for the satellite developer.

The limit checker becomes even more "limited" when this test setting is scaled up into reality. With the redundancy and the custom thresholds we included in our limit checker watchpoint table, we already used up 51 out of 176 possible watchpoints for the default cFS watchpoint table size, or almost a third of watchpoint slots. This does not yet include watchpoints for any other subsystem, such as the power system or thermals, both of which require careful monitoring so as to not lead to physical destruction of the satellite. Thus, not only is using the limit checker to handle possible security issues burdensome on the developer, there may be size, power, and runtime constraints on constructing a more carefully-tuned limit checker.

4.7 Part I Summary

In this chapter part, we demonstrated that common existing spaceflight software platforms are ill-equipped to handle malicious satellite peripherals. In particular, we demonstrated a dilemma for space systems in which either the system deploys countermeasures against malicious components and suffers potentially unacceptable nominal performance, or the

system cannot survive malicious components. We show this along two axes: manipulation of data and manipulation of time, and show that these axes of attacks avoid detection while leading to unacceptable slowdown or mission failure. We demonstrated this satellite peripheral dilemma on the core Flight System from NASA in a state-of-the-art emulated, closed-loop environment.

Chapter 4 Part II: Addressing the Compromised Peripheral Dilemma

4.8 Additional Motivation

Many cyber-physical systems (CPS) underpin essential infrastructure, including power grids, chemical processing plants, agriculture, and nuclear facilities. CPS also form the foundation of much emerging technology, including space systems and robotics. As CPS continue to increase in importance and complexity, the likelihood of attacks against them only increases. Previous attacks on cyber-physical infrastructure have already demonstrated severe physical and financial impacts, such as the Stuxnet attack targeting nuclear centrifuges [46], the German steel mill attack causing extensive physical damage [95], or the irreparable damage to sun sensors on the ROSAT satellite that has been speculatively attributed to cybersecurity incidents [126]. And the recent discovery of the fast16 malware targeting demonstrates the decades-long attention on targeting CPS and their simulation software [77].

Since Stuxnet, the security community has made progress on understanding and improving CPS security, including for cars [86], improving security of satellite flight computers and communication links [75, 160], security for drones and other robotic vehicles [34, 35], medical devices [62], and even DNA sequencers [118].

One threat model receiving insufficient attention is a high-privileged compromised peripheral, where a sensor is compromised either via a firmware trojan or through exploiting some vulnerability later in development. This malicious periphal can send data at any value or time period. In particular, there is an increasing risk of blackbox COTS devices making

their way onto important systems, such as space vehicles, industrial control systems, and other critical infrastructure. Thus, CPS operators must trust that the output is honest without knowing the inner workings of the device. We explore potential defenses to this threat in this paper, with a focus on satellites specifically.

Previous work has made strides towards securing the controllers of CPS (the main software portion that commands actuators and receives sensor input), such as improving the flight computer component of satellites [55]. However, there has been less attention on defending against malicious peripheral attacks, especially when the attacker is assumed to be persistent. Chapter 4 Part I evaluated the impact of such a persistent attack on a satellite; Chapter Part II aims to fill in the gap of finding an onboard defense towards such an attack.

The increasing risk of compromised peripherals stems from CPS industry trends. Many industries, including the space industry, have seen more distribution in suppliers of third-party commercial-off-the-shelf components. Many of these COTS components are blackbox and thus cannot be fully verified for lack of security vulnerabilities or malicious firmware. In addition, the development cycles of these cyber-physical systems continue to speed up, allowing even less time and lowering incentives for companies to do in-depth supply chain validation for each of their products. Thus, there is a need for defense in-depth, where peripherals are assumed to be possibly malicious or otherwise faulty.

Sensor diagnosis, also known as sensor attribution, is an emerging area of system security research aimed at identifying malicious peripherals on a CPS [36, 54]. Our work aims to find effective sensor diagnosis techniques against malicious peripherals.

Our contributions are the following:

- We motivate a compromised peripheral threat model and establish characteristics of systems that might benefit from an automated sensor diagnosis technique
- We propose ControlPhoenix, a novel sensor diagnosis technique. We instantiate ControlPhoenix in an ADCS control loop within an extensive, high-fidelity satellite simulation. ControlPhoenix uses mission requirements to detect possible attacks,

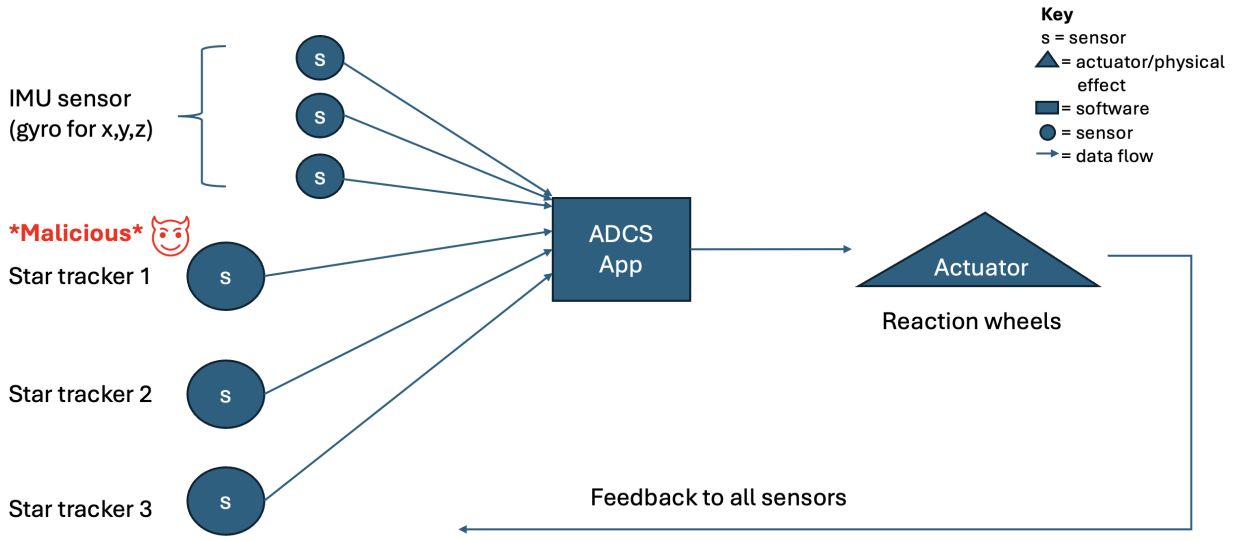


Figure 4.2: Threat model of a malicious peripheral on a satellite. This is the attitude determination and control subsystem (ADCS). In this case, the only malicious component is the star tracker peripheral. Data flows from the sensors into the sensor fusion algorithm component, which leads to the move command, which leads to attitude maneuvers of the satellite by its reaction wheels actuators. Since this control algorithm is a combination of software computation and physical action, the malicious peripheral data may slow down the change in orientation movement, lead to the wrong orientation, or possible other negative effects.

wherein approaching a violation of mission requirements triggers the attack diagnosis procedure.

4.9 Problem Description & Threat Model

In this section we describe in-depth the compromised peripheral attack. We first describe a motivating scenario that would require sensor diagnosis, explain what the threat model is and what the malicious component capabilities are, then enumerate what assumptions a

CPS should meet to be a candidate for sensor diagnosis.

4.9.1 Motivating Scenario

Suppose a state adversary wants to target a fleet of satellites with shared common hardware and software, such as a fleet of internet satellites, or a company that sells satellites at scale to many different government customers. Both these company models are popular today. Such a high-resourced adversary may seek to initiate space warfare, for example causing satellite fleets to hit other objects and achieve cascading debris effects, or hit other manmade space objects and blame the mistake on whoever owns the satellite, as warned in previous work [111]. Or, in the case of satellites that facilitate ground communications or internet use, an adversary may want to take a timed attack that cuts off important communication infrastructure all at once. Ample previous work has enumerated possible threats to a compromised satellite [32, 43, 45, 73, 83, 84, 103, 111, 126, 126], and these threats are only multiplied when targeting more than one satellite at once.

4.9.2 Threat Source and Capabilities

We consider a supply chain attack on a CPS with a persistent hold in a blackbox peripheral. This peripheral, or set of compromised peripherals, exists within a larger control system in a CPS. See Figure 4.2 for an example compromised peripheral on a satellite.

A malicious peripheral may have access to real sensor measurements, and can send whatever sensor measurements they want from the peripheral at any frequency. Peripherals are assumed to send only malicious data or take malicious action without compromising any other parts of a system. For example, a satellite flight computer can meet security properties, but the sensor values it takes in could lead to incorrect actions, given that the flight computer assumes that the sensor readings are honest or faulty, not malicious.

In addition, we assume that the adversary knows the anomaly detection and/or recovery algorithms in the system, but does not have access to individual defense system parameters that are specific to a satellite mission.

4.9.3 CPS Assumptions

In order to be a candidate for sensor diagnosis in the threat model of a compromised peripheral, the CPS must meet the following set of assumptions.

1. The CPS is made up of one or more closed-loop control systems. These are loops of sensors, controllers, and actuators achieving physical effects that do not require human intervention. See Section 4.10 for a more detailed description and figure.
2. There is peripheral redundancy. For any sensor diagnosis approach to work with this threat model, the misbehaving sensor(s) need to be able to be shut off permanently, or temporarily if an honest fault. If there are multiple malicious peripherals, there must be backup components that enable the CPS to continue its behavior, even if at degraded quality. In the case of a satellite, having both a compromised star tracker and a compromised IMU sensor does not necessarily spell mission failure, as many advanced satellites are equipped with a copy of the IMU, and there is also typically attitude sensor redundancy.
3. Honest sensors/supply chain diversity. Complementing Assumption 2, any compromised peripheral defense would not work if each sensor in a redundant set is malicious. There must exist honest components at each segment of the control loop, even if the majority of peripherals are compromised. To decrease the likelihood that all sensors are malicious, it would be preferable to purchase different COTS components from different vendors. Per Figure 4.2, the satellite developer would ideally purchase their attitude sensors from at least two different vendors.
4. Any automated sensor diagnosis approach is most useful when the CPS cannot be completely shut off when an anomaly occurs. Either the system is difficult or impossible to access, such as a satellite in orbit, or the system needs to maintain availability, such as a power grid or for financial reasons. In absence of the possibility for immediate and accurate human intervention, an automated sensor diagnosis system is needed to identify and respond to anomalies.

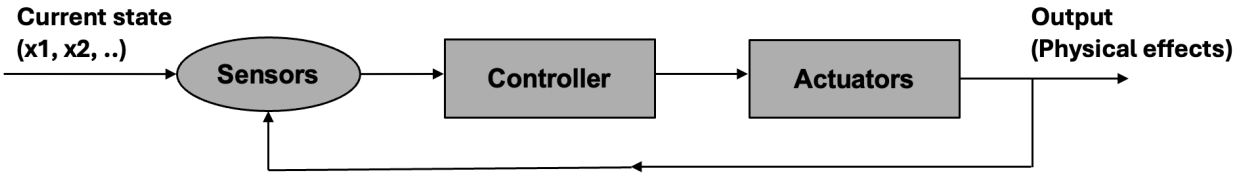


Figure 4.3: Control system at a high level. Sensors generate an estimate of the current state, then the controller commands an actuator to take action based on that state.

5. A set of measurable cyber-physical requirements for the CPS is an important component of our specific sensor diagnosis approach. We describe further the reason for this in the design section, but the CPS needs to have a set list of requirements that, if violated, do not meet safety, availability, or reliability constraints. As described further below, many CPS already have this, including satellites.

4.10 Background

Control Systems. Control systems, as shown in Figure 4.3, take in some inputs through sensors, process them with a controller, and produce some physical outputs [69]. For example, the attitude determination and control system (ADCS) on a satellite takes in measured orientation data from sensors as input, determines the orientation based on "fusing" sensor data from multiple sensors, then outputs some physical movement. Or, consider a satellite's thermal control system where the inputs are heat from the environment and from onboard processing, and the outputs are temperature adjustments, such as dissipating heat in warm environments or heating up systems in cold environments. Both of these systems are considered closed-loop control systems, meaning that based on inputs, the system can adjust itself to a desired state. Control systems typically have time requirements. Either there is a hard requirement, and getting delayed can damage the system and cause physical harm, or there is a soft requirement, and slowdowns to the expected control flow can cause degraded performance.

Satellite System Architecture. A satellite is comprised of multiple control systems, including ADCS and thermals. The main "brain" of the satellite is the flight computer (acting as the controller across multiple control systems on the satellite). The flight computer interfaces with multiple peripheral hardware components, which are typically connected to the flight computer through a serial link.

Satellites traditionally emphasize resiliency in order to maintain a multi-year long mission without any physical access. Traditional resiliency techniques include sensor redundancy and a limit checker. Sensor redundancy means there are multiple sensors on board the satellite that aim to measure the same property, and one getting damaged should not impact the ability to complete the mission. A limit checker is a tool that monitors flight computer messages to check that the values are in preconfigured threshold ranges. If a value exceeds a threshold, the limit checker may take some automated action. Chapter 4 Part I showed that these existing resiliency techniques are insufficient for defending against a compromised peripheral attack.

Sensor Attacks & Detection. Past work has shown that during sensor fusion, a malicious sensor can achieve devastating physical effects by evading sensor smoothing patterns and triggering an incorrect state estimation [12,61,87,99,100,102,170,171]. Other prior work has attempted detection and mitigation techniques to prevent malicious sensors from attacking systems using Kalman filters [25, 47]. One such mitigation technique is to evaluate the difference between a sensor reading and the model-predicted state, where the model is often a Kalman filter. However, this relies on the model of the system and the sensor itself being very accurate, which is often not the case, particularly in nonlinear systems like satellite attitude determination or thermals. And prior work shows that Kalman filters can be evaded with stealthy attacks and lead to drifting of state over time.

Park et al. proposed pairwise sensor comparisons across redundant sensors, though they note that their detection system does not take into account stealthy attacks [125]. Pairwise comparisons are one check in a toolbox that can be used for sensor diagnosis, though its

impact may be limited when different sensor types are fused together or if more than one sensor is compromised.

Urbina et al. summarize work on industrial-control system attacks, including stealthy attacks where attackers have full control of sensors and/or actuators [154]. They use physics models to tightly control the sorts of attacks that stealthy attackers can do. This requires that the physics models be sufficiently accurate, which is much more difficult in complex, nonlinear systems such as thermal control systems or satellite attitude determination where the sensors themselves do not operate under perfect conditions (for example star tracker camera sensors being impacted by sun or radiation). Chapter 4 Part I showed that tightly-controlled anomaly detection thresholds could not support nominal, non-attack cases, leading to a dilemma in whether to set thresholds to handle malicious peripherals or nominal mission conditions [113].

Satellite Fault Detection. Many satellites have some form of fault isolation, detection, and recovery (FIDR) mechanism on board [66, 168]. To detect sensor faults, systems may use a Sensor Built-in Test (BIT), where the sensor itself reports whether a fault may be occurring. This would not be effective against a malicious peripheral given that it is in control of the output of the BIT. Another approach is the pairwise sensor comparison mentioned above, which would not be effective necessarily in stealthy attacks, diverse sensor types, or when the majority of sensors are malicious. Operators may also use hard-coded values to detect a fault [66], such as tracking the temperature and seeing if it goes out of range, though this still does not handle the diagnosis portion of which sensors or actuators are malicious. Lastly, traditional fault detection in satellites may use residual analysis, where the sensor measurements are thrown out if they exceed some threshold of difference predicted by the mathematical state model [66, 168]. However, as described above, this is evaded via stealthy attacks, and runs into the limitation of requiring a sufficiently accurate state model.

Sensor Attack Recovery & Sensor Diagnosis. Previous recovery techniques assume sensor readings are being attacked from physical channels and that they are recoverable from such

attacks [27, 36]. Our work considers an expanded threat model, where the attack can be persistent, which means it can perform long-term, stealthy attacks, and must be shut off completely for the CPS to recover from any attack impacts. Unlike other work, we also assume there is sensor redundancy as this is the norm for many complex CPS. PID-Piper is effective on robotic vehicles (RV's) due to its ability to accurately model the RV's direction control loop [35]. Similarly, DeLorean diagnoses sensors through factor graph calculations, where likelihood of a sensor being malicious is tied to the difference between expected state and measured state [36]. These solutions run into the same challenge as other attack detection and recovery solutions listed above, as it relies on a highly accurate state model.

Lu et al. summarize existing recovery techniques for CPS, calling for more work on sensor diagnosis and more recovery work for specific domains, such as healthcare [101]. Our work on sensor diagnosis for CPS, starting specifically with the satellite domain, aims to address this gap.

4.11 ControlPhoenix Design

A new recovery and sensor diagnosis technique is needed for CPS enduring a long-term malicious peripheral. We propose ControlPhoenix, an approach for sensor attribution that handles many cases for malicious peripheral attacks on CPS.

Below, we list a set of design requirements that a sensor diagnosis system would need to meet given our threat model and CPS assumptions from Section 4.9.3. Then, we propose ControlPhoenix, a generic model for sensor diagnosis that handles many cases for malicious peripheral attacks on CPS, with special attention to the needs of satellites.

4.11.1 Design requirements

1. Lightweight: Limited Compute/Realtime Constraints. The amount of memory and power for a sensor attribution system is limited. Sensor diagnosis on a satellite should ideally be able to fit on a moderately complex satellite in the 100kg - 500 kg range

without violating any hard or soft control loop time constraints.

2. Minimal Developer overhead: In order for a sensor diagnosis approach to be adapted by industry, developer overhead should be minimal. They should not have to spend much time doing any types of model training or parameter tuning.
3. Chapter 3 of this dissertation found evidence that reliability in satellites is valued over security [111]. The approach should not have excessive false positives that hamper nominal, non-malicious conditions.
4. Handles multiple types of peripheral attacks: It is not possible to anticipate what peripheral or what type of attack an adversary will employ. Thus, the solution should assume that any sensor or actuator on the CPS is a possible malicious target, and not anticipate what malicious values an attacker might send, or malicious actuations an attacker might take.

4.11.2 Generic Design

In this section, we describe the general approach for ControlPhoenix. In the section after, we show a concrete of ControlPhoenix instantiation on a satellite simulation.

ControlPhoenix consists of the following steps:

1. A health monitor continuously runs and analyzes data across the CPS. The health monitor makes use of specific CPS mission requirements and sub-requirements, and these are used as rules to test whether in range, out of range, or outside of the safety margin.
2. If a health monitor metric is out of range (or close to out of range), the attack diagnosis step is activated.
3. Attack diagnosis. Thus far, there is only knowledge that a requirement is violated, not the source peripheral(s). The attack diagnosis step runs a series of randomized functional actuation-based tests to identify malfunctioning sensors by ignoring the measurements of one or a subset of sensors at a time. By evaluating these test results

on key metrics like runtime and power usage and other health metrics, ControlPhoenix diagnoses which periperhals cause the requirement deviations.

System Requirements Health Monitor

The first portion of ControlPhoenix is the health monitor. The health monitor checks over a set list of thresholds specified by the CPS developer in a table. The table can be up to size N, dependent on the CPS constraints. There are up to N requirements specified in the health monitor. Each requirement must have a range specified of both a minimum and maximum value, along with a percent safety margin. If there is only a minimum or only a maximum value, this can also be configured. Then, if a value exceeds a margin but is still within the requirement range, sensor diagnosis starts. An example rule that could exist on a satellite is "thermals shall ensure that the on-board computers shall be maintained within their operating temperature range $-X$ to Y°C." If the temperature reading goes into a margin close to $-X$ or Y, the sensor diagnosis will start turning relevant peripherals off and on.

Peripheral Representation

In order to run the sensor diagnosis algorithm, there needs to be a way to know which peripherals are candidates for being malicious or faulty. Luckily, the structure of control loops themselves allow for some level of separation between requirements and control loops. See Figure 4.4 for an example with satellites. If a temperature requirement is violated, the sensor diagnosis algorithm will first test only the peripherals involved in the thermal subsystem, not the ADCS loop. This allows the problem of sensor diagnosis for a complex system to be achievable, as increasing complexity of a CPS might not lead to explosion of state in the number of peripherals to test during diagnosis since each requirement will not connect to every single control loop (but every control loop should have a requirement that monitors it).

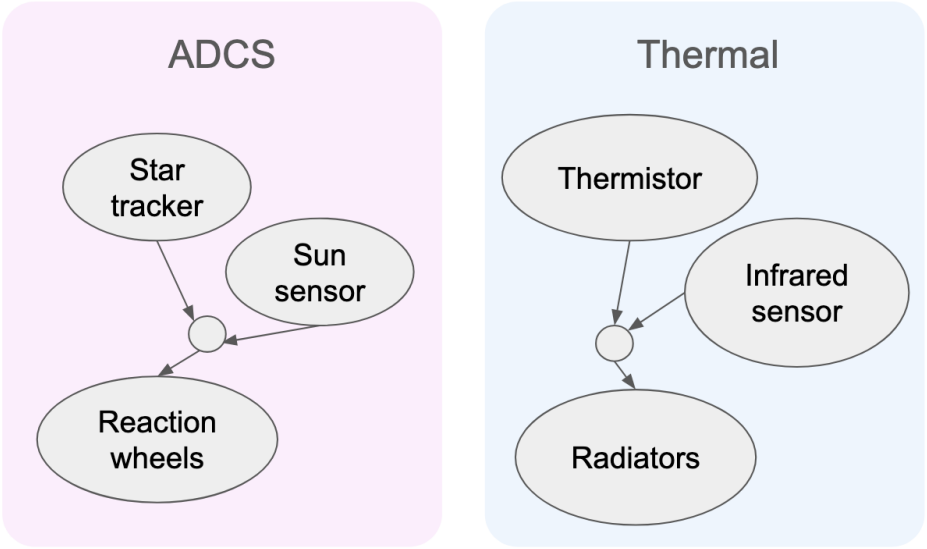


Figure 4.4: Control systems within a CPS can be seen as disconnected graphs. If a requirement is violated, it is typically linked to a specific control system, from which the problem of searching for malicious peripherals can be constrained.

Diagnosis Algorithm

There are many possible approaches to the choice of which peripherals to turn off first, and for how long. Some of these parameters depend on the physical properties of a control system. For example, in the satellite scenario below, when turning a sensor off, we check that within a short period the requirement starts trending back inward, then take further action if it starts to maintain its requirements again.

4.12 ControlPhoenix on a Satellite

4.12.1 Implementation

We instantiate and evaluate ControlPhoenix on a representative satellite mission from the perspective of a malicious star tracker in its ADCS control loop. The ADCS control loop is an essential part of all satellite functionality, as it controls how the satellite is oriented.

This ensures that the payload, radio, and possibly solar arrays are pointed properly. If the satellite is not pointed properly, or it begins to spin uncontrollably, i.e., tumble, the satellite could become unrecoverable and the mission could fail.

We focus on satellite ADCS (compared to other control systems on a satellite) for multiple reasons: 1) it is possible to simulate with sufficient fidelity to evaluate our tool without having to test it on a real satellite, 2) ADCS has existing public research papers that give insight into common practices in industry and government, and 3) there exists well-developed open-source tooling to simulate astrodynamics in space [78].

The mission represents an ADCS loop that uses three redundant star trackers, one IMU (one gyro each on the x,y, and z axis), a Kalman filter that ingests IMU and star tracker data to output an attitude estimate, and then reaction wheel simulations that represent the attitude actuators.

We simulate our peripherals using the Basilisk astrodynamics simulation framework [78], specifically a star tracker for estimating attitude and an IMU to simulate gyroscopes in the x,y, and z axis for angular velocity. These peripheral simulations are ingested into the satellite simulation, which we built using NASA cFS, a common flight software framework used across industry and government. cFS is built on top of Linux. We built and incorporated multiple applications atop cFS, including a star tracker application, IMU application, limit checker for the health monitor portion of ControlPhoenix, a reaction wheel application, and an ADCS application for taking in attitude sensors and sending off a command to the reaction wheels on how they should move. We use three redundant star trackers, an IMU, and simulate reaction wheels. See Figure 4.2 for the control loop we use.

The requirement rules that we use in ControlPhoenix are translated directly from real satellite design documentation into paramaters appropriate for our simulation. We also use an industry-standard extended Kalman filter for attitude determination logic, as well as use industry-realistic peripheral setups of star trackers, an IMU, and reaction wheels. Lastly, we simulate a camera payload, inspired by examples of real satellite payloads. The camera payload is intended to be a "cheaper" version of the star tracker, and also less accurate. We

also simulate this with Basilisk, but add higher noise than the actual ADCS loop.

Another aspect of simulation realism is that the simulation enables two separate modes for the satellite: experiment mode and nominal mode. Experiment mode is when the payload is being tested, which will be the bulk of the mission. Experiment mode should hold at a specific orientation the whole time so that the experimental star tracker payload it can take test images of the star field. Nominal mode is when the satellite is orbiting and not in experiment mode. In nominal mode, the orientation may not be still, and the satellite can maneuver from any orientation A to any orientation B.

Attitude Estimation: Kalman Filter. To estimate attitude, most satellites in industry use an Extended Kalman Filter (EKF). We derive our EKF parameters and dimensions from NASA [96], but made choices on individual parameters based on our simulation design, such as the amount of star tracker noise we added to the simulation. The derivation of our EKF is extensive. We include a more thorough description in Appendix C.2. To handle multiple star trackers at once, we use an adapted version of the sensor fusion technique called sequential filtering [164]. All EKF code was written as a C library and integrated into the NASA cFS ADCS application.

4.12.2 Satellite Health Monitor

Our ControlPhoenix instantiation makes use of the NASA cFS limit checker as a tool to monitor messages on the cFS software bus. We adapt four requirements from real satellite documentation across multiple anonymized satellites. It is industry-standard to design satellites around mission requirements, so every satellite has requirements for every subsystem (and thus each control system on the satellite). These are all real requirements for ADCS subsystems. The requirements are adapted to be as follows:

1. **Attitude stability $<$ 1 deg/s in experiment mode.** The reason the small satellite with the camera payload needs this requirement is that this prevents the camera payload from taking blurry photos in experiment mode. If the angular rate gets close to

this through an attack, something might be wrong. This is tracked by having the limit checker listen to the IMU data and check the angular rate for each gyro on the x,y,z axis. Angular stability is calculated as $\sqrt{g_x^2 + g_y^2 + g_z^2}$. After some parameter testing, we also derived an attitude stability metric for nominal mode: **Attitude stability $<$ 13 deg/s in nominal mode.**

2. The bus shall have point stability relative to payload deck: X arcseconds per y milliseconds at $Z\sigma$ in experiment mode. This requirement is more about short-term jitter, and is useful for high-precision payloads. This requirement uses a rolling window jitter metric based on $Z\sigma$. This requires measuring the random variation in the y milliseconds time window. Then, compute the standard deviation σ and check that $Z * \sigma \leq X$ arcsec.

   Because star tracker data only sends at 1hz, this requirement uses IMU jitter as a proxy since it comes in at 10hz. After some experimentation around what amounts of jitter showed up holding at the origin both with and without attacks, the following parameters made most sense to use for this simulation:

   The bus shall have point stability relative to payload deck: 7000 arcseconds per 10 seconds at $3Z\sigma$ in experiment mode.
3. **Cameras pointed $\geq 70\,\text{deg}$ away from sun when cameras on (in experiment mode).** This requirement prevents damage of cameras and enables the payload camera to actually collect its target data without being blocked by the sun. Each test in evaluation assumes a constant sun location relative to the payload body, so the sun location is hardcoded in and then compared to the current orientation estimated by the extended Kalman filter.
4. **The payload star tracker attitude shall agree with the ADCS attitude estimate within 20 degrees.** The camera payload, which is a novel star tracker sensor, should be within 20 deg of its target attitude.

The requirement values are then forwarded to our custom cFS ControlPhoenix application for diagnosis logic and checking in with the health monitor thresholds. ControlPhoenix listens on the software bus for requirement messages from the limit checker, then refers to a health monitor table in the ControlPhoenix app directory. Each health monitor table element has the requirement id, lower limit, upper limit, lower margin, upper margin, and flag for whether the requirement check is enabled.

4.12.3 Diagnosis Algorithm Parameters

Based on observation prior to running all functional tests, we decided on the following sensor diagnosis approach for the satellite simulation:

- Requirement Triggered: A requirement exceeds the safety margin, so ControlPhoenix checks that it exceeds that margin for three more data points. If it doesn't, it goes back to `idle` mode. If the requirement exceeds the safety margin for 3 seconds or more, a peripheral in the ADCS loop at random is chosen and ControlPhoenix enters `try_sensor` mode.
- Try sensor: The evaluation currently chooses from three star trackers. The candidate star tracker is turned off for 20 seconds. If within those 20 seconds, the trend of the requirement begins to go inward through a sliding-window least squares test of 5 data points, the candidate star tracker enters `recovery` mode.
- Attempt Recovery: In `recovery` mode, ControlPhoenix waits for 45 seconds to see whether 5 straight data points exist where *all* requirements for the ADCS control loop are met. If they are all met for 5 straight points, ControlPhoenix enters the `confirm` mode.
- Confirm Malicious Sensor: To be confirmed as a malicious sensor, all ADCS requirements must be met for 30 seconds straight. If they are not, ControlPhoenix picks another sensor and goes back to `try_sensor` mode for that sensor.

4.13 ControlPhoenix Satellite Evaluation

Table 4.11: Honest Scenario: ControlPhoenix running when no attack is occuring. Results averaged across 6 trials, where each trial is a permutation of the ordering of star trackers being turned off for testing. AUC results are shown alongside the percentage of trials in which each star tracker was correctly identified, accounting for both false positives and false negatives. Nominal mode of jitter only and camera-sun angle only are N/A because no such requirement exists in nominal mode.

Enabled requirements	**Experiment mode (still)**				**Nominal mode (moving)**			
	AUC	**Correctly identified**			**AUC**	**Correctly identified**		
		ST1	**ST2**	**ST3**		**ST1**	**ST2**	**ST3**
No requirements enabled	0.053	100%	100%	100%	3.672	100%	100%	100%
Jitter only	0.052	100%	100%	100%	N/A	N/A	N/A	N/A
Payload accuracy only	0.052	100%	100%	100%	3.671	100%	100%	100%
Attitude stability only	0.054	100%	100%	100%	3.671	100%	100%	100%
Camera-sun angle only	0.055	100%	100%	100%	N/A	N/A	N/A	N/A
All requirements enabled	0.052	100%	100%	100%	3.671	100%	100%	100%

4.13.1 Experimental Setup

We run a set of functional tests on the ADCS in both experimental and nominal mode, with and without ControlPhoenix turned on. We show that ControlPhoenix defends against multiple types of attacks that previously evaded traditional satellite resiliency techniques [113]. The experiment mode test attempts to hold at a fixed orientation for four

Table 4.12: Data attack type 1: Non-stealthy attack. Random quaternions between -1 and 1 (in proper quaternion range) at each timestep, then normalized. Results averaged across 6 trials, where each trial is a permutation of the ordering of star trackers being turned off for testing. AUC results are shown alongside the percentage of trials in which each star tracker was correctly identified, accounting for both false positives and false negatives. Nominal mode of jitter only and camera-sun angle only are N/A because no such requirement exists in nominal mode.

Enabled requirements	**Experiment mode (still)**				**Nominal mode (moving)**			
	AUC	**Correctly identified**			**AUC**	**Correctly identified**		
		ST1	**ST2**	**ST3**		**ST1**	**ST2**	**ST3**
No requirements enabled	26.252	0%	100%	100%	29.203	0%	100%	100%
Jitter only	26.665	100%	100%	100%	N/A	N/A	N/A	N/A
Payload accuracy only	27.410	100%	100%	100%	30.911	100%	100%	100%
Attitude stability only	27.986	100%	100%	100%	34.777	0%	94.444%	88.889%
Camera-sun angle only	21.593	100%	100%	100%	N/A	N/A	N/A	N/A
All requirements enabled	28.378	100%	100%	100%	35.561	72.222%	100%	100%

Table 4.13: Data attack type 2. Semi-stealthy attack. Malicious star tracker increases its quaternion values by 0.3 at each timestep, then normalized.

Enabled requirements	**Experiment mode (still)**				**Nominal mode (moving)**			
	AUC	**Correctly identified**			**AUC**	**Correctly identified**		
		ST1	**ST2**	**ST3**		**ST1**	**ST2**	**ST3**
No requirements enabled	31.534	0%	100%	100%	32.057	0%	100%	100%
Jitter only	61.956	100%	100%	100%	N/A	N/A	N/A	N/A
Payload accuracy only	69.851	83.333%	100%	100%	49.153	88.889%	100%	100%
Attitude stability only	60.591	100%	100%	100%	38.481	22.222%	100%	100%
Camera-sun angle only	59.851	100%	100%	100%	N/A	N/A	N/A	N/A
All requirements enabled	65.651	83.333%	100%	100%	58.489	66.667%	100%	100%

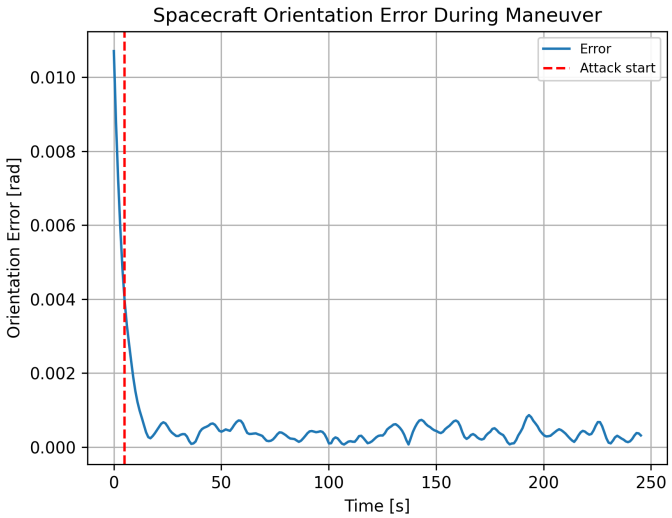


Figure 4.5: Time versus attitude error in honest scenario in experiment mode. CP is never triggered and the star tracker holds close to the origin.

Table 4.14: Data attack type 3. Stealthy attack. Malicious star tracker increases its quaternion values by 0.05 at each timestep, then normalized.

Enabled requirements	**Experiment mode (still)**				**Nominal mode (moving)**			
	AUC	**Correctly identified**			**AUC**	**Correctly identified**		
		ST1	**ST2**	**ST3**		**ST1**	**ST2**	**ST3**
No requirements enabled	139.074	0%	100%	100%	134.355	0%	100%	100%
Jitter only	90.717	100%	100%	100%	N/A	N/A	N/A	N/A
Payload accuracy only	90.434	100%	100%	100%	105.787	88.889%	100%	100%
Attitude stability only	91.347	100%	100%	100%	109.264	88.889%	100%	100%
Camera-sun angle only	50.374	100%	100%	100%	N/A	N/A	N/A	N/A
All requirements enabled	90.273	100%	100%	100%	104.205	83.333%	100%	100%

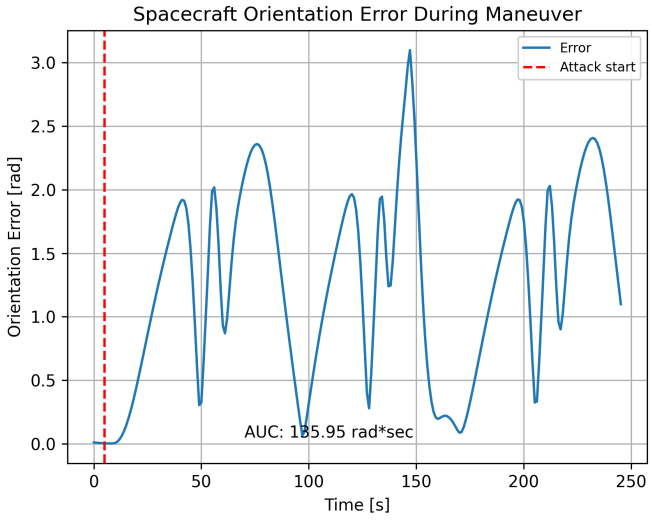


Figure 4.6: Stealthy attack with no defenses.

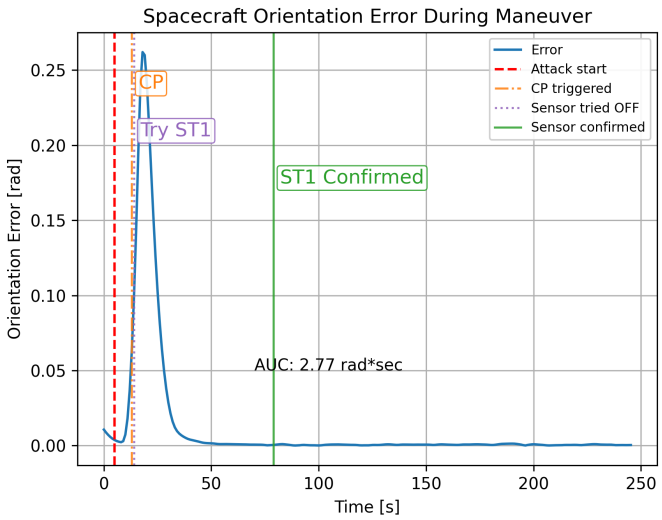


Figure 4.7: Stealthy attack with malicious star tracker (ST1) being turned off first.

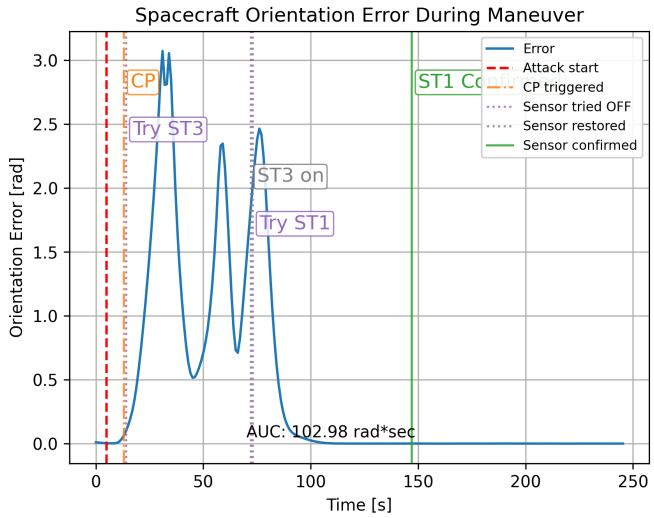


Figure 4.8: Stealthy attack with malicious star tracker (ST1) being turned off second.

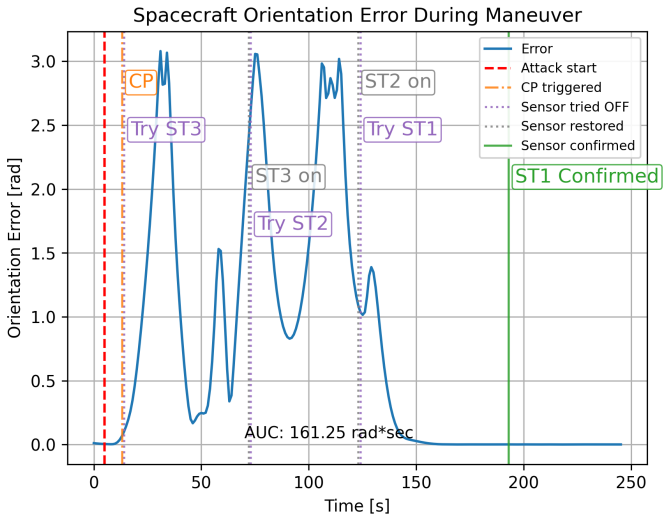


Figure 4.9: Stealthy attack with malicious star tracker (ST1) being turned off third.

minutes, in this case the origin of $q_0 = [1, 0, 0, 0]^T$. The nominal mode tests three different attitude maneuvers, where the starting attitude is the origin and the target attitudes were chosen at random:

$q_1 = [0.822, -0.329, 0.329, 0.329]^T$

$q_2 = [0.928, 0.216, 0.216, 0.216]^T$

$q_3 = [0.600, 0.462, -0.462, -0.462]^T$.

4.13.2 Honest Scenario

We first test ControlPhoenix in an honest scenario to confirm that it does not cause any false positives. See Table 4.11 for more details. Across all trials with the mission-specific parameters, ControlPhoenix never triggers in an honest scenario. This is useful for adoption, as the satellite industry values reliability more than security [111].

4.13.3 Non-stealthy Attack

We next evaluate the first attack type, which sends data completely at random at each timestamp, yet still within range of a quaternion attitude measure (normalized and with each quaternion between -1 and 1). Despite being completely random data, the AUC values tended to be lower than those in the stealthier test. This is likely a feature of the extended Kalman filter, as the trust value of the sensors goes down the further away it is from the model's expectation. Therefore, the model component of the filter is relied on higher, which works as a smoothing approach in simulation but will be slightly less effective in real-world noisy scenarios. Still, the malicious star tracker is detected correctly always in experiment mode, which is what mode the satellite would be in the bulk of its mission time. See Table 4.12.

4.13.4 Semi-stealthy Attack

We evaluted a semi-stealthy attack type aimed at drifting the sensor away from its truth value but not in a way as extreme as truly random values. Malicious star tracker increases its quaternion values by 0.3 at each timestep, then normalized. See Table 4.13 for detailed results.

4.13.5 Stealthy Attack

We evaluted a stealthy attack type aimed at drifting the sensor away from its truth value in a way that varies only slightly each timestamp. Malicious star tracker increases its quaternion values by 0.05 at each timestep, then normalized. See Table 4.14 for detailed results.

4.14 Part II Summary

In this paper, we proposed ControlPhoenix, a novel approach to sensor diagnosis for persistent peripheral attacks in cyber-physical systems. We engineered an instantiation of ControlPhoenix on a satellite attidue determination and control loop, and showed that

malicious peripherals were mostly correctly identified after attacking. We hope this serves as a foundation for further work on sensor diagnosis that can be adapted in the satellite industry and beyond.

Chapter 5
Conclusion

Satellites form an unofficial category of critical infrastructure, and this makes them an important technology to consider from a security perspective. While there has been increased attention on satellite security in the academic community prior to and concurrent with this dissertation, gaps remained. These gaps included understanding the practical barriers and challenges to protecting the satellite ecosystem, considering threat models that either did not exist in previous work or were not thoroughly explored, and finding solutions to the aforementioned threats, barriers, and challenges.

To better understand practical barriers and challenges, this dissertation used empirically grounded methodology. This includes surveys in Chapter 2 on understanding concerns and threats to individual privacy of commercial satellites, in-depth interviews and code analysis in Chapter 3 to understand barriers university smallsat developers face, and experimentally evaluating compromised peripheral threats along with a defense in Chapter 4.

To work towards filling any possible gaps in threat models, this dissertation considered threat models in all chapters that have previously received little attention. In Chapter 2, this dissertation brought attention to the threat model of a malicious non-government entity using satellite imagery to infringe upon individual privacy. In Chapter 3, this dissertation explored the possible threats of and to amateur smallsats, and how even a smallsat can damage the satellite ecosystem. Chapter 4 considered a compromised peripheral on a complex satellite.

Lastly, this dissertation made steps towards finding solutions to many of the threats identified in this dissertation. Chapter 2 posed possible mitigations to privacy harms from the perspective of regulators, legislators, and developers. Chapter 3 discussed possible solutions to the barriers identified earlier in the chapter in order to improve smallsat security. And

Chapter 4 Part II proposed and experimentally evaluated ControlPhoenix, a way to overcome the malicious peripheral dilemma identified in Chapter 4 Part I. Together, this dissertation works towards better protecting the satellite ecosystem.

Bibliography

[1] Annual Number of Objects Launched Into Space. Our World in Data, Aug. 2023. https://ourworldindata.org/space-exploration-satellites.

[2] Abbasi, A., Wetzels, J., Holz, T., and Etalle, S. Challenges in Designing Exploit Mitigations for Deeply Embedded Systems. In *2019 IEEE European Symposium on Security and Privacy (EuroS&P)* (2019), IEEE, pp. 31–46.

[3] Abdalla, M., Bellare, M., Catalano, D., Kiltz, E., Kohno, T., Lange, T., Malone-Lee, J., Neven, G., Paillier, P., and Shi, H. Searchable Encryption Revisited: Consistency Properties, Relation to Anonymous IBE, and Extensions. In *Advances in Cryptology – CRYPTO* (Aug. 2005).

[4] Adilov, N., Alexander, P., Cunningham, B., and Albertson, N. An Analysis of Launch Cost Reductions for Low Earth Orbit Satellites. *Economics Bulletin* (2022).

[5] Ae Chun, S., and Atluri, V. Protecting Privacy from Continuous High-Resolution Satellite Surveillance. *Data and Application Security: Developments and Directions* (2001), 233–244.

[6] Alleven, M. EarthNow Satellite Venture Backed by Bill Gates, SoftBank Emerges Out of Stealth Mode. Fierce Wireless, Apr. 2018. https://www.fiercewireless.com/wireless/earthnow-satellite-venture-backed-by-bill-gates-softbank-emerges-out-stealth-mode.

[7] Alvarez Leon, L. F. An Emerging Satellite Ecosystem and the Changing Political Economy of Remote Sensing. In *The Nature of Data: Infrastructures, Environments, Politics*, J. Goldstein and E. Nost, Eds. U of Nebraska Press, 2022, pp. 71–102.

[8] AMSAT. AMSAT - The Radio Amateur Satellite Corporation. https://www.amsat.org/.

[9] Anik, M. T. H., Guilley, S., Danger, J.-L., and Karimi, N. On the Effect of Aging on Digital Sensors. In *2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID)* (2020), IEEE, pp. 189–194.

[10] Atluri, V., and Chun, S. A. An Authorization Model for Geospatial Data. *IEEE Transactions on Dependable and Secure Computing 1*, 4 (2004), 238–254.

[11] Attitude, A., and Systems, H. R. Extended kalman filter. `https://ahrs.readthedocs.io/en/latest/index.html`.

[12] Bai, C.-Z., Gupta, V., and Pasqualetti, F. On Kalman Filtering With Compromised Sensors: Attack Stealthiness and Performance Bounds. *IEEE Transactions on Automatic Control 62*, 12 (2017), 6641–6648.

[13] Bartel Sheehan, K. An Investigation of Gender Differences in On-Line Privacy Concerns and Resultant Behaviors. *Journal of Interactive Marketing 13*, 4 (1999), 24–38.

[14] Beam, C. Soon, Satellites Will Be Able to Watch You Everywhere All the Time. MIT Technology Review, June 2019. `https://www.technologyreview.com/2019/06/26/102931/satellites-threaten-privacy/`.

[15] Bennett, M., Van Den Hoek, J., Zhao, B., and Prishchepov, A. Improving Satellite Monitoring of Armed Conflicts. *Earth's Future 10*, 9 (2022), e2022EF002904.

[16] Bennett, M. M. Chinese Sociotechnical Imaginaries of Earth Observation: From Sight to Foresight. *Big Data & Society 10*, 2 (2023), 20539517231191527.

[17] Bertelsen, O., Bødker, S., Eriksson, E., Hoggan, E., and Vermeulen, J. Beyond Generalization: Research for the Very Particular. *ACM Interactions* (2019).

[18] Black, J. T. Commercial Satellites: Future Threats or Allies? *Naval War College Review 52*, 1 (1999), 99–114.

[19] Borisov, N., Brumley, D., Wang, H. J., Dunagan, J., Joshi, P., Guo, C., et al. Generic Application-Level Protocol Analyzer and Its Language. In *NDSS* (2007).

[20] Braun, V., and Clarke, V. Using Thematic Analysis in Psychology. *Qualitative Research in Psychology 3*, 2 (2006), 77–101.

[21] Breeden, A. French Tax Collectors Use A.I. to Spot Thousands of Undeclared Pools. The New York Times, Aug. 2022. `https://www.nytimes.com/2022/08/30/world/europe/france-taxes-pools-artificial-intelligence.html`.

[22] Brunetti, G., Campiti, G., Tagliente, M., and Ciminelli, C. COTS Devices for Space Missions in LEO. *IEEE Access* (2024).

[23] CCSDS. The Consultative Committee for Space Data Systems. https://public.ccsds.org.

[24] Chang, V., Chundury, P., and Chetty, M. Spiders in the Sky: User Perceptions of Drones, Privacy, and Security. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems* (2017), pp. 6765–6776.

[25] Chang, Y. H., Hu, Q., and Tomlin, C. J. Secure Estimation Based Kalman Filter for Cyber–Physical Systems Against Sensor Attacks. *Automatica 95* (2018), 399–412.

[26] Chin, Jamie and Coelho, Roland and Foley, Justin and Johnstone, Alicia and Nugent, Ryan and Pignatelli, Dave and Pignatelli, Savannah and Powell, Nikolaus and Puig-Suari, Jordi. *CubeSat 101: Basic Concepts and Processes for First-Time CubeSat Developers*. NASA, 2017. https://www.nasa.gov/wp-content/uploads/2017/03/nasa_csli_cubesat_101_508.pdf?emrc=05d3e2.

[27] Choi, H., Kate, S., Aafer, Y., Zhang, X., and Xu, D. Software-based Realtime Recovery from Sensor Attacks on Robotic Vehicles. In *International Symposium on Recent Advances in Intrusion Detection* (2020).

[28] Cohen, J. A Coefficient of Agreement for Nominal Scales. *Educational and Psychological Measurement* (1960).

[29] CONFERS. Comments of the Consortium for the Execution of Rendezvous and Servicing Operations. CONFERS, 2019. https://www.satelliteconfers.org/wp-content/uploads/2019/07/CONFERS_Comment_NOAA_NPRM_07152019.pdf.

[30] CONFERS. CONFERS Homepage. CONFERS, 2020. https://www.satelliteconfers.org/.

[31] Curtmola, R., Garay, J., Kamara, S., and Ostrovsky, R. Searchable Symmetric Encryption: Improved Definitions and Efficient Constructions. In *ACM Conference on Computer and Communications Security* (2006).

[32] Cyr, B., Long, Y., Sugawara, T., and Fu, K. Position Paper: Space System Threat Models Must Account for Satellite Sensor Spoofing. In *SpaceSec* (2023).

[33] Daniel, B. K. Using the TACT Framework to Learn the Principles of Rigour in Qualitative Research. *Electronic Journal of Business Research Methods* (2019).

[34] Dash, P., Chan, E., and Pattabiraman, K. SpecGuard: Specification Aware Recovery for Robotic Autonomous Vehicles from Physical Attacks. *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security* (2024).

[35] Dash, P., Li, G., Chen, Z., Karimibiuki, M., and Pattabiraman, K. PID-Piper: Recovering Robotic Vehicles from Physical Attacks. In *51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks* (2021), IEEE.

[36] Dash, P., Li, G., Karimibiuki, M., and Pattabiraman, K. Diagnosis-Guided Attack Recovery for Securing Robotic Vehicles from Sensor Deception Attacks. ASIA CCS '24, Association for Computing Machinery, p. 915–929.

[37] de Selding, P. B. Airbus Sells In-orbit Spot 7 Imaging Satellite to Azerbaijan. SpaceNews, Dec 2014.

[38] Denning, T. The Security Cards: A Security Threat Brainstorming Toolkit, 2013. `https://securitycards.cs.washington.edu/`.

[39] Devaraj, K., Kingsbury, R., Ligon, M., Breu, J., Vittaldev, V., Klofas, B., Yeon, P., and Colton, K. Dove High Speed Downlink System. *Smallsat* (2017).

[40] Ecoffet, R. Overview of In-Orbit Radiation Induced Spacecraft Anomalies. *IEEE Transactions on Nuclear Science 60*, 3 (2013), 1791–1815.

[41] Engineering, G., and Technology Directory, N. core Flight System, 2025. `https://etd.gsfc.nasa.gov/capabilities/core-flight-system/`.

[42] Engineering, G., and Technology Directory, N. core Flight System Limit Checker Application Version 2, 2025. `https://software.nasa.gov/software/GSC-16010-1`.

[43] Falco, G. The Vacuum of Space Cyber Security. In *2018 AIAA SPACE and Astronautics Forum and Exposition* (2018).

[44] Falco, G., and Boschetti, N. A Security Risk Taxonomy for Commercial Space Missions. In *ASCEND 2021*. 2021, p. 4241.

[45] Falco, G., Viswanathan, A., and Santangelo, A. CubeSat Security Attack Tree Analysis. In *IEEE SMC-IT* (2021).

[46] Falliere, N., Murchu, L. O., Chien, E., et al. W32.Stuxnet Dossier. *White Paper, Symantec Corp., Security Response 5*, 6 (2011).

[47] Fang, H., Haile, M. A., and Wang, Y. Robust Extended Kalman Filtering for Systems With Measurement Outliers. *IEEE Transactions on Control Systems Technology 30*, 2 (2021), 795–802.

[48] Federal Communications Commission. 47 CFR 97.211 - Space Station Control Operator License Grant Eligibility. *Code of Federal Regulations, Title 47 Part 97*. https://www.ecfr.gov/current/title-47/section-97.211.

[49] Flores, A., and Belongie, S. Removing Pedestrians from Google Street View Images. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition-Workshops* (2010), IEEE, pp. 53–58.

[50] Flores-Anderson, A. I., Herndon, K. E., Thapa, R. B., and Cherrington, E. The SAR Handbook: Comprehensive Methodologies for Forest Monitoring and Biomass Estimation. Tech. rep., 2019.

[51] Formeller, C. Introducing 15 cm HD: The Highest Clarity from Commercial Satellite Imagery. Maxar, Nov. 2020. https://blog.maxar.com/earth-intelligence/2020/introducing-15-cm-hd-the-highest-clarity-from-commercial-satellite-imagery.

[52] FOX 5 DC Digital Team. Thomas Jefferson High School Students Launch Satellite on SpaceX Rocket. Fox 5 Washington DC, Nov. 2022. https://www.fox5dc.com/news/thomas-jefferson-high-school-students-launch-satellite-on-spacex-rocket.

[53] Frome, A., Cheung, G., Abdulkader, A., Zennaro, M., Wu, B., Bissacco, A., Adam, H., Neven, H., and Vincent, L. Large-Scale Privacy Protection in Google Street View. In *2009 IEEE 12th International Conference on Computer Vision* (2009), IEEE, pp. 2373–2380.

[54] Fung, C., Zeng, E., and Bauer, L. Attributions for ML-based ICS Anomaly Detection: From Theory to Practice. *Proceedings 2024 Network and Distributed System Security Symposium* (2024).

[55] Furgala, J., Jero, S., Lin, A., and Skowyra, R. Porting NASA's core Flight System to the Formally Verified seL4 Microkernel. In *Workshop on the Security of Space and Satellite Systems (SpaceSec 2026)* (2026).

[56] Galesic, M., and Tourangeau, R. What Is Sexual Harassment? It Depends on Who Asks! Framing Effects on Survey Responses. *Applied Cognitive Psychology: The Official Journal of the Society for Applied Research in Memory and Cognition* (2007).

[57] Giuliari, G., Ciussani, T., Perrig, A., and Singla, A. ICARUS: Attacking Low Earth Orbit Satellite Networks. In *USENIX ATC* (2021).

[58] Golej, P., Orlikova, L., Horak, J., Linhartova, P., and Struhar, J. Detection of People and Vehicles Using Very High-resolution Satellite Images. *International Multidisciplinary Scientific GeoConference: SGEM 21*, 2.1 (2021), 283–291.

[59] Goodchild, M., Appelbaum, R., Crampton, J., Herbert, W., Janowicz, K., Kwan, M.-P., Michael, K., Alvarez León, L., Bennett, M., Cole, D. G., et al. A White Paper on Locational Information and the Public Interest.

[60] Google. Blur or Remove 360 Imagery & Photo Paths. Google, 2023. `https://support.google.com/maps/answer/7011973`.

[61] Hallyburton, R. S., Khazraei, A., and Pajic, M. Optimal Myopic Attacks on Nonlinear Estimation. In *2022 IEEE 61st Conference on Decision and Control (CDC)* (2022), IEEE, pp. 5480–5485.

[62] Halperin, D., Benjamin, T., Fu, K., Kohno, T., and Maisel, W. Security and Privacy for Implantable Medical Devices. *IEEE Pervasive Computing 7* (02 2008), 30–39.

[63] Halperin, D., Heydt-Benjamin, T. S., Ransford, B., Clark, S. S., Defend, B., Morgan, W., Fu, K., Kohno, T., and Maisel, W. H. Pacemakers and Implantable Cardiac Defibrillators: Software Radio Attacks and Zero-Power Defenses. In *2008 IEEE S & P* (2008), IEEE.

[64] Hiniker, A., Hong, S., Kohno, T., and Kientz, J. A. MyTime: Designing and Evaluating an Intervention for Smartphone Non-Use. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (2016), pp. 4746–4757.

[65] Hitchens, T. NOAA Eases Licensing Restrictions on Commercial Remote Sensing. Breaking Defense, Aug. 2023. `https://breakingdefense.com/2023/08/noaa-eases-licensing-restrictions-on-commercial-remote-sensing`.

[66] Hogan, S. L. Effective Fault Management Guidelines. Tech. rep., The Aerospace Corporation, June 2009.

[67] House, T. W. FACT SHEET: Vice President Harris Advances National Security Norms in Space. The White House, Apr. 2022. https://www.whitehouse.gov/briefing-room/statements-releases/2022/04/18/fact-sheet-vice-president-harris-advances-national-security-norms-in-space/.

[68] in Data, O. W. Cost of Space Launches to Low Earth Orbit. https://ourworldindata.org/grapher/cost-space-launches-low-earth-orbit.

[69] Institute, N. S. S. S. V. Space Vehicle Control Systems. Tech. rep., NASA, 2023.

[70] Iridium. Iridium Satellite Time & Location. https://www.iridium.com/satellite-time-location/.

[71] Janowicz, K., Gao, S., McKenzie, G., Hu, Y., and Bhaduri, B. GeoAI: Spatially Explicit Artificial Intelligence Techniques for Geographic Knowledge Discovery and Beyond, 2020.

[72] Jean, N., Burke, M., Xie, M., Davis, W. M., Lobell, D. B., and Ermon, S. Combining Satellite Imagery and Machine Learning to Predict Poverty. *Science 353*, 6301 (2016), 790–794.

[73] Jedermann, E., Strohmeier, M., Lenders, V., and Schmitt, J. RECORD: A RECeption-Only Region Determination Attack on LEO Satellite Users. In *USENIX Security* (2024).

[74] Jedermann, E., Strohmeier, M., Schäfer, M., Schmitt, J., and Lenders, V. Orbit-Based Authentication Using TDOA Signatures in Satellite Networks. In *Proceedings of the 14th ACM Conference on Security and Privacy in Wireless and Mobile Networks* (2021).

[75] Jero, S., Furgala, J., Heller, M. A., Nahill, B., Mergendahl, S., and Skowyra, R. Securing the Satellite Software Stack. In *SpaceSec* (2024).

[76] Jewett, R. NOAA Eliminates Certain Restrictions on Commercial Remote Sensing Companies. Via Satellite, Aug 2023.

[77] Kamluk, V., and Andrés Guerrero-Saade, J. fast16 — Mystery Shadow Brokers Reference Reveals High-Precision Software Sabotage 5 Years Before Stuxnet . Sentinel Labs, Apr. 2026. https://www.sentinelone.com/labs/fast16-mystery-s

hadowbrokers-reference-reveals-high-precision-software-sabotage-5-years-before-stuxnet/?ref=zetter-zeroday.com.

[78] Kenneally, P. W., Piggott, S., and Schaub, H. Basilisk: A Flexible, Scalable and Modular Astrodynamics Simulation Framework. *Journal of Aerospace Information Systems 17*, 9 (2020).

[79] Kerlin, S. D., Straub, J., Huhn, J., and Lewis, A. Small Satellite Communications Security and Student Learning in the Development of Ground Station Software. In *IEEE Aerospace Conference* (2015).

[80] Kessler, D. J., and Cour-Palais, B. G. Collision Frequency of Artificial Satellites: The Creation of a Debris Belt. *Journal of Geophysical Research: Space Physics* (1978).

[81] Kessler, D. J., Johnson, N. L., Liou, J., and Matney, M. The Kessler Syndrome: Implications to Future Space Operations. *Advances in the Astronautical Sciences* (2010).

[82] Kimball, R. Using NDVI Differences to Measure Drought in the Russian River Watershed. Planet, July 2021. https://www.planet.com/pulse/using-ndvi-differences-to-measure-drought-in-the-russian-river-watershed/.

[83] Koisser, D., Mitev, R., Chilese, M., and Sadeghi, A. Don't Shoot the Messenger: Localization Prevention of Satellite Internet Users. In *IEEE S&P* (2024).

[84] Koisser, D., Mitev, R., Yadav, N., Vollmer, F., and Sadeghi, A.-R. Orbital Trust and Privacy: SoK on PKI and Location Privacy Challenges in Space Networks. In *USENIX Security* (2024).

[85] Korda, M. Widespread Blurring of Satellite Images Reveals Secret Facilities . Federation of American Scientists, Dec. 2018. https://fas.org/publication/widespread-blurring-of-satellite-images-reveals-secret-facilities/.

[86] Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., McCoy, D., Kantor, B., Anderson, D., Shacham, H., et al. Experimental Security Analysis of a Modern Automobile. In *2010 S & P* (2010), IEEE.

[87] Kung, E., Dey, S., and Shi, L. The Performance and Limitations of ϵ-Stealthy Attacks on Higher Order Systems. *IEEE Transactions on Automatic Control 62*, 2 (2016), 941–947.

[88] Kurzrok, A., Ramos, M. D., and Mechentel, F. Evaluating the Risk Posed by Propulsive Small Satellites With Unencrypted Communications Channels to High-Value Orbital Regimes. *Smallsat* (2018).

[89] La Rosa, L. Recent String of Celebrity Home Burglaries Linked to Social Media Posts: Police. ABC News, oct 2018. `https://abcnews.go.com/Entertainment/recent-string-celebrity-home-burglaries-linked-social-media/story`.

[90] Labs, P. Planet Labs Homepage. `https://www.planet.com/`.

[91] Lane, D., Leon, E., Solio, D., Cunningham, D., Obukhov, D., and Tacliad, F. C. High-Assurance Cyber Space Systems for Small Satellite Mission Integrity. *Smallsat* (2017).

[92] Lazar, J., Feng, J. H., and Hochheiser, H. *Research Methods in Human-Computer Interaction*. Morgan Kaufmann, 2017.

[93] League, A. R. R. Amateur Radio Emergency Service - When All Else Fails. `https://www.arrl.org/ares`.

[94] Leary, M. G. The Missed Opportunity of United States v. Jones: Commercial Erosion of Fourth Amendment Protection in a Post-Google Earth World. *U. Pa. J. Const. L. 15* (2012), 331.

[95] Lee, R. M., Assante, M. J., and Conway, T. German Steel Mill Cyber Attack. *Industrial Control Systems 30*, 62 (2014), 1–15.

[96] Lefferts, E. J., Markley, F. L., and Shuster, M. D. Kalman Filtering for Spacecraft Attitude Estimation. *Journal of Guidance, Control, and Dynamics 5*, 5 (1982), 417–429.

[97] Lewis, J. J., and Caplan, L. R. Drones to Satellites: Should Commercial Aerial Data Collection Regulations Differ by Altitude? *SciTech Lawyer 11*, 4 (2015), 10.

[98] Li, A., Wang, J., and Zhang, N. Software Availability Protection in Cyber-Physical Systems. In *34th USENIX Security Symposium* (2025).

[99] Liu, Y., Xue, W., He, S., and Cheng, L. Stealthy False Data Injection Attacks Against Extended Kalman Filter Detection in Power Grids. In *2021 8th International Conference on Information, Cybernetics, and Computational Social Systems (ICCSS)* (2021), IEEE, pp. 459–464.

[100] Lu, J., and Niu, R. A State Estimation and Malicious Attack Game in Multi-Sensor Dynamic Systems. In *2015 18th International Conference on Information Fusion (Fusion)* (2015), IEEE, pp. 932–936.

[101] Lu, P., Zhang, L., Liu, M., Sridhar, K., Sokolsky, O., Kong, F., and Lee, I. Recovery from Adversarial Attacks in Cyber-physical Systems: Shallow, Deep, and Exploratory Works. *ACM Computing Surveys* (Apr. 2024).

[102] Ma, Y., Sharp, J. A., Wang, R., Fernandes, E., and Zhu, X. Sequential Attacks on Kalman Filter-Based Forward Collision Warning Systems. In *Proceedings of the AAAI Conference on Artificial Intelligence* (2021), vol. 35, pp. 8865–8873.

[103] Manulis, M., Bridges, C. P., Harrison, R., Sekar, V., and Davis, A. Cyber Security in New Space: Analysis of Threats, Key Enabling Technologies and Challenges. *International Journal of Information Security* (2021).

[104] Maral, G., Bousquet, M., and Sun, Z. *Satellite Communications Systems: Systems, Techniques and Technology*. John Wiley & Sons, 2020.

[105] Mathews, L. Researchers Used a Decommissioned Satellite to Broadcast Hacker TV. Wired, Mar. 2022. `https://www.wired.com/story/satellite-hacking-anit-f1r-shadytel/`.

[106] Mathews, L. Viasat Reveals How Russian Hackers Knocked Thousands of Ukrainians Offline. Forbes, Mar. 2022. `https://www.forbes.com/sites/leemathews/2022/03/31/viasat-reveals-how-russian-hackers-knocked-thousands-of-ukrainians-offline/`.

[107] Maurice-Michel, D. How I Hacked an ESA's Experimental Satellite. deadf00d. `https://www.deadf00d.com/post/how-to-hack-an-esa-experimental-satellite.html`.

[108] Maxar. Next-Generation Weather Satellite Technology. `https://www.maxar.com/splash/weather`.

[109] Maxar. What Does It Take to Make WorldView Legion? Maxar, 2023.

[110] McAllister, E., Payo, A., Novellino, A., Dolphin, T., and Medina-Lopez, E. Multispectral Satellite Imagery and Machine Learning for the Extraction of Shoreline Indicators. *Coastal Engineering* (2022), 104102.

[111] McAmis, R., Haas, G., Sim, M., Kohlbrenner, D., and Kohno, T. Short: Unencrypted Flying Objects: Security Lessons from University Small Satellite Developers and Their Code. In *Symposium on Vehicle Security and Privacy (VehicleSec'25)* (2025).

[112] McAmis, R., Sim, M., Bennett, M., and Kohno, T. Over Fences and Into Yards: Privacy Threats and Concerns of Commercial Satellites. *PoPETS* (2024).

[113] McAmis, R., Willison, C., Skowyra, R., and Mergendahl, S. The Compromised Satellite Peripheral Dilemma.

[114] McKenna, A. T., Gaudion, A. C., and Evans, J. L. The Role of Satellites and Smart Devices: Data Surprises and Security, Privacy, and Regulatory Challenges. *Penn St. L. Rev. 123* (2018), 591.

[115] Mead, N. R., Shull, F., Vemuru, K., and Villadsen, O. A Hybrid Threat Modeling Method. *CMU/SEI-2018-TN-002* (2018).

[116] Morin, J. Is Space Becoming the Next Front for War—And Traffic Jams? Yale Insights. https://insights.som.yale.edu/insights/is-space-becoming-the-next-front-for-war-and-traffic-jams.

[117] NASA. CubeSat Launch Initiative. NASA. https://www.nasa.gov/kennedy/launch-services-program/cubesat-launch-initiative/.

[118] Ney, P. M., Koscher, K., Organick, L., Ceze, L., and Kohno, T. Computer Security, Privacy, and DNA Sequencing: Compromising Computers With Synthesized DNA, Privacy Leaks, and More. In *USENIX Security Symposium* (2017).

[119] Nodari, A., Vanetti, M., and Gallo, I. Digital Privacy: Replacing Pedestrians from Google Street View Images. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)* (2012), IEEE, pp. 2889–2893.

[120] Nsoesie, E. O., Rader, B., Barnoon, Y. L., Goodwin, L., and Brownstein, J. Analysis of Hospital Traffic and Search Engine Data in Wuhan China Indicates Early Disease Activity in the Fall of 2019.

[121] O'Callaghan, J. Trump Accidentally Revealed the Amazing Resolution Of U.S. Spy Satellites. Forbes, Sept. 2019. https://www.forbes.com/sites/jonathanocallaghan/2019/09/01/trump-accidentally-revealed-the-amazing-resolution-of-u-s-spy-satellites/?sh=17b78f4b3d89.

[122] Oceanic, N., and Administration, A. Licensing of Private Remote Sensing Space Systems. Federal Register, May 2020. https://www.federalregister.gov/documents/2020/05/20/2020-10703/licensing-of-private-remote-sensing-space-systems.

[123] of State, U. S. D. International Traffic in Arms Regulations. https://www.ecfr.gov/current/title-22/chapter-I/subchapter-M.

[124] Parcak, S. Are We Ready for Satellites That See Our Every Move? The New York Times, Oct. 2019. https://www.nytimes.com/2019/10/15/opinion/satellite-image-surveillance-that-could-see-you-and-your-coffee-mug.html.

[125] Park, J., Ivanov, R., Weimer, J., Pajic, M., and Lee, I. Sensor Attack Detection in the Presence of Transient Faults. In *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems* (2015).

[126] Pavur, J., and Martinovic, I. Building a Launchpad for Satellite Cyber-Security Research: Lessons from 60 Years of Spaceflight. *Journal of Cybersecurity* (2022).

[127] Planet. How Does This Compare to Imagery in Applications Like Google Maps? Planet, April 2017. https://support.planet.com/hc/en-us/articles/115005684627-How-does-this-compare-to-imagery-in-applications-like-Google-Maps-.

[128] Planet. High-Resolution Imagery With Planet Satellite Tasking. Planet, 2023. https://www.planet.com/products/hi-res-monitoring/.

[129] Planet. Norway's International Climate and Forests Initiative Satellite Data Program. Planet, 2023. https://www.planet.com/nicfi/.

[130] Planet. Overview. Planet, May 2023. https://developers.planet.com/docs/planetschool/satellite-imagery/.

[131] Planet. Planet. Planet, 2023. https://www.planet.com/.

[132] Planet. SkySat. Planet, May 2023. https://developers.planet.com/docs/data/skysat/.

[133] Prybylo, M., Haghighi, S., Peddinti, S. T., and Ghanavati, S. Evaluating Privacy Perceptions, Experience, and Behavior of Software Development Teams. In *Twentieth Symposium on Usable Privacy and Security (SOUPS 2024)* (Aug. 2024), USENIX Association.

[134] Purdy, R. Ruling on Sharper Satellite Images Poses a Privacy Problem We Can No Longer Ignore. The Conversation, June 2014. https://theconversation.com/ruling-on-sharper-satellite-images-poses-a-privacy-problem-we-can-no-longer-ignore-28133.

[135] Ramulu, H. S., Schmitt, H., Wermke, D., and Acar, Y. Security and Privacy Software Creators' Perspectives on Unintended Consequences. In *USENIX Security* (2024).

[136] Redmiles, E. M., Acar, Y., Fahl, S., and Mazurek, M. L. A Summary of Survey Methodology Best Practices for Security and Privacy Researchers, 2017.

[137] Saha, S. S., Rahman, S., Ahmed, M. U., and Aditya, S. K. Ensuring Cybersecure Telemetry and Telecommand in Small Satellites: Recent Trends and Empirical Propositions. *IEEE Aerospace and Electronic Systems Magazine* (2019).

[138] Sarah Peters. Middle School Students' Satellite Launched Into Space. The Daytona Beach News-Journal, Dec. 2018. https://www.news-journalonline.com/story/news/state/2018/12/31/middle-school-students-satellite-launched-into-space/6414842007/.

[139] SatNOGS. SatNOGS – Open Source Global Network of Satellite Ground-Stations. https://satnogs.org/.

[140] Schaeffer, K. Among U.S. Couples, Women Do More Cooking and Grocery Shopping Than Men. Pew Research, Sept. 2019. https://www.pewresearch.org/short-reads/2019/09/24/among-u-s-couples-women-do-more-cooking-and-grocery-shopping-than-men/.

[141] Schneider, D. U.S. Eases Restrictions on Private Remote-Sensing Satellites. IEEE Spectrum, July 2020. https://spectrum.ieee.org/eased-restrictions-on-commercial-remote-sensing-satellites.

[142] Shepherd, J. D., Schindler, J., and Dymond, J. R. Automated Mosaicking of Sentinel-2 Satellite Imagery. *Remote Sensing 12*, 22 (2020), 3680.

[143] Simko, L., Chin, B., Na, S., Saluja, H. K., Zhu, T. Q., Kohno, T., Hiniker, A., Yip, J., and Cobb, C. Would You Rather: A Focus Group Method for Eliciting and Discussing Formative Design Insights with Children. In *Interaction Design and Children* (2021), pp. 131–146.

[144] Simmon, R. Earth at a Cute Angle. Medium, Jan. 2021. https://medium.com/nightingale/earth-at-a-cute-angle-dc2f8c29495a.

[145] SkyFi. SkyFi Features. SkyFi, 2023. https://www.skyfi.com/features.

[146] Song, D. X., Wagner, D., and Perrig, A. Practical Techniques for Searches on Encrypted Data. In *IEEE Symposium on Security and Privacy* (2000).

[147] Steele, L. J. The View from on High: Satellite Remote Sensing Technology and the Fourth Amendment. *High Tech. LJ 6* (1991), 317.

[148] Swartwout, M. Reliving 24 Years in the Next 12 Minutes: A Statistical and Personal History of University-Class Satellites. *Smallsat* (2018).

[149] Sweeney, L. Simple Demographics Often Identify People Uniquely. *Health (San Francisco) 671*, 2000 (2000), 1–34.

[150] Tedeschi, P., Sciancalepore, S., and Di Pietro, R. Satellite-Based Communications Security: A Survey of Threats, Solutions, and Research Challenges. *Computer Networks 216* (2022).

[151] The European Space Agency. SPOT. The European Space Agency, 2023.

[152] Turow, J., Hennessy, M., Draper, N., Akanbi, O., and Virgilio, D. Divided We Feel: Partisan Politics Drive Americans' Emotions Regarding Surveillance of Low-Income Populations. *A Report from the Annenberg School for Communication University of Pennsylvania* (2018).

[153] United States Space Force. Defense Support Program Satellites. https://www.spaceforce.mil/about-us/fact-sheets/article/2197774/defense-support-program-satellites/, 2020.

[154] Urbina, D. I., Giraldo, J. A., Cardenas, A. A., Tippenhauer, N. O., Valente, J., Faisal, M., Ruths, J., Candell, R., and Sandberg, H. Limiting the Impact of Stealthy Attacks on Industrial Control Systems. In *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security* (New York, NY, USA, 2016), CCS '16, Association for Computing Machinery, p. 1092–1105.

[155] USGS. Number of Commercial, Government-Civil Satellites Launched, 2025. https://www.usgs.gov/media/images/number-commercial-government-civil-satellites-launched.

[156] Usman, M., Qaraqe, M., Asghar, M. R., and Shafique Ansari, I. Mitigating Distributed Denial of Service Attacks in Satellite Networks. *Transactions on Emerging Telecommunications Technologies* (2020).

[157] Vedda, J. A. Updating National Policy on Commercial Remote Sensing. *Center for Space Policy and Strategy. The Aerospace Corporation//http://www. aerospace. org/wp-content/uploads/2017/03/CommercialRemoteSensing. 21* (2017), 2018.

[158] Wang, D., Szymanski, B. K., Abdelzaher, T., Ji, H., and Kaplan, L. The Age of Social Sensing. *Computer 52*, 1 (2019), 36–45.

[159] Wang, Y., Xia, H., Yao, Y., and Huang, Y. Flying Eyes and Hidden Controllers: A Qualitative Study of People's Privacy Perceptions of Civilian Drones in the US. *Proc. Priv. Enhancing Technol. 2016*, 3 (2016), 172–190.

[160] Werner, D. NASA to Roll Out Major Update to core Flight Software, Feb 2025. `https://spacenews.com/nasa-to-roll-out-major-update-to-core-flight-software/`.

[161] Wertz, J. R., Ed. *Spacecraft Attitude Determination and Control*, vol. 73 of *Astrophysics and Space Science Library*. Springer Netherlands, Dordrecht, 1978.

[162] Willbold, J., Schloegel, M., Göhler, F., Scharnowski, T., Bars, N., Wörner, S., Schiller, N., and Holz, T. Scaling Software Security Analysis to Satellites: Automated Fuzz Testing and Its Unique Challenges. In *IEEE Aerospace Conference* (2024).

[163] Willbold, J., Schloegel, M., Vögele, M., Gerhardt, M., Holz, T., and Abbasi, A. Space Odyssey: An Experimental Software Security Analysis of Satellites. In *IEEE S&P* (2023).

[164] Willner, D., Chang, C.-B., and Dunn, K.-P. Kalman Filter Algorithms for a Multi-Sensor System. In *1976 IEEE Conference on Decision and Control Including the 15th Symposium on Adaptive Processes* (1976), IEEE, pp. 570–574.

[165] Wong, M. Y., Landen, M., Li, F., Monrose, F., and Ahamad, M. Comparing Malware Evasion Theory With Practice: Results from Interviews With Expert Analysts. In *SOUPS* (2024).

[166] Wulder, M. A., and Coops, N. C. Satellites: Make Earth Observations Open Access. *Nature 513*, 7516 (2014), 30–31.

[167] Xu, W., Chen, X., Anderson, M., Drager, S., and Kong, F. Recovery-Guaranteed Sensor Attack Detection for Cyber-Physical Systems . In *2025 IEEE 31st Real-Time and Embedded Technology and Applications Symposium (RTAS)* (2025).

[168] Yin, S., Xiao, B., Ding, S. X., and Zhou, D. A Review on Recent Development of Spacecraft Attitude Fault Tolerant Control System. *IEEE Transactions on Industrial Electronics* (2016).

[169] Zeng, E., Mare, S., and Roesner, F. End User Security and Privacy Concerns With Smart Homes. In *SOUPS* (2017).

[170] Zhang, Z., Zhou, L., and Tokekar, P. Strategies to Inject Spoofed Measurement Data to Mislead Kalman Filter. *arXiv preprint arXiv:1710.02442* (2017).

[171] Zhu, Y., Miao, C., Xue, H., Yu, Y., Su, L., and Qiao, C. Malicious Attacks Against Multi-Sensor Fusion in Autonomous Driving. In *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking* (2024), pp. 436–451.

[172] Zisk, R. SkyFi Launches Satellite Imagery App. Payload, Jan. 2023. `https://payloadspace.com/skyfi-launches-satellite-imagery-app/`.

Appendix A

Chapter 2

A.1 Survey Questions

[Consent Form] We are researchers at the University of Washington (UW) studying the relationship between people and satellites in space. This study was reviewed by the UW Institutional Review Board (IRB) and deemed exempt because it involves no more than minimal risk and meets other criteria. Your responses to this survey will be anonymized. Data from this survey will be stored securely and kept confidential. Your participation in this study is voluntary. You may withdraw your participation at any time. If you have questions about this study, you may contact the PI at satellites@cs.washington.edu. You may also contact the UW Human Subjects Division (HSD), which manages IRB review, at hsdinfo@uw.edu.

I am at least 18 years old, I have read and understood this consent form, and I agree to participate in this online research study. ∘ Yes ∘ No [End survey if no]

Are you an employee of the University of Washington, family member of a UW employee, or UW student involved in this particular research? ∘ Yes ∘ No [End survey if yes]

[Filtering] For this survey, we require participants to be above a certain level of corrected vision and computer screen quality, since some later questions involve evaluating images. How many people are there in the image below? (Answer in the empty box below the picture) [Photo shown is Figure A.1] _____ [End survey if answer is not 5]

Figure A.1: Photo we show to respondents to filter for vision impairments

[Introduction] This study has 8 sections and we estimate it will take 20 minutes or less

[Filters for definition comprehension] In the following questions, we will consider imagery taken from directly above (perpendicular to) the ground, such as what satellites in outer space might capture. We define this as the "satellite view". We list some examples below of images that are "satellite view" and images that are not "satellite views". This is a satellite view, since the image is taken directly above (perpendicular to) the ground: [Survey includes example of an image taken from satellite view] This is not a satellite view: [Survey includes example of an image taken *not* from satellite view]

Is the image below a "satellite view" image? (To move on to the remaining questions, you must answer this question correctly.) [Image shown that is not a satellite view] ∘ Yes ∘ No [End survey if yes]

Is the image below a "satellite view" image? (To move on to the remaining questions, you must answer this question correctly.) [Image shown that is a satellite view] ∘ Yes ∘ No [End survey if no]

Is the image below a "satellite view" image? (To move on to the remaining questions,

you must answer this question correctly.) [Image shown that is not a satellite view] ∘ Yes ∘ No [End survey if yes]

[Awareness and Perception] Commercial satellite companies collect images from satellites in space and sell these images to non-governmental individuals, non-governmental organizations, and governments for a variety of applications, including agriculture, defense, and scientific research. Individuals, such as you, are allowed to purchase these images.

Before this survey, had you heard of commercial satellites selling images of the Earth to non-governmental individuals and organizations? ∘ Yes ∘ No

Do you see any ways that the use of commercial satellite images could be beneficial? Please list them below. (If you cannot think of any, please enter "None"). [Free-response text] _____

Do you see any ways that the use of commercial satellite images could cause harms? Please list them below. (If you cannot think of any, please enter "None"). [Free-response text] _____

Below is the same image of a parking lot but at different resolutions. Which of the following images do you think represents the highest-resolution commercial satellite camera when the satellite is taking the image from space? ∘ [Image of cars in parking lot downsampled to 10m resolution] ∘ [Image of cars in parking lot downsampled to 3m resolution] ∘ [Image of cars in parking lot downsampled to 1m resolution] ∘ [Image of cars in parking lot downsampled to 15cm resolution]

Suppose someone wants to see how the parking lot from the previous question changes across time. How often do you think a single commercial satellite company can take an image of the same parking lot? ∘ Every year ∘ Every month ∘ Every week ∘ Every day ∘ Every hour ∘ Every minute ∘ Every second

How much do you think it costs for an individual to buy one satellite image (in US dollars)? For reference, one satellite image could be over 10 square miles of the earth. [Number] _____

[Scenarios] Satellite images have a variety of capabilities to capture different activities, places, and people. Below, you will see pairs of hypothetical satellite capabilities. For each pair, decide whether you would prefer satellites to have both capabilities, or whether you would prefer satellites to have neither of these capabilities.

Consider scenarios A and B. A: Satellite images are used to track the number of cars at a hospital, which helps with predicting levels of disease in the area. B: Someone calls in sick to work. To check whether they are at home, their employer takes a satellite image over their home and observes whether their car is there. Would you rather satellites have both of these capabilities, or neither of these capabilities? ∘ Both scenarios should be allowed ∘ Neither scenario should be allowed

Why did you choose your answer to the previous question? [Text box] _____

Consider scenarios C and D. C: Someone wants to know whether their former romantic partner is having any visitors to their house, and uses a satellite image to detect how many people are gathered in their former romantic partner's backyard. D: Someone's backyard catches fire, and their insurance provider uses a satellite image to confirm the items that were lost due to the fire, which ends up giving the fire victim more money than expected. Would you rather satellites have both of these capabilities, or neither of these capabilities? ∘ Both scenarios should be allowed ∘ Neither scenario should be allowed

Why did you choose your answer to the previous question? [Text box] _____

Consider Scenarios E and F. E: Satellite images are used to detect what a well-known celebrity is doing in their backyard. F: Satellite images are used to locate a hiker that went missing in the mountains. Would you rather satellites have both of these capabilities, or neither of these capabilities? ∘ Both scenarios should be allowed ∘ Neither scenario should be allowed

Why did you choose your answer to the previous question? [Text box] _____

Consider scenarios G and H. G: Satellite images are used to detect and find endangered species based on tracks the species leave on the ground, which helps with conservation efforts. H: Satellite images are used to detect and find endangered species, based on tracks the species leave on the ground, which people use to illegally hunt the endangered species. Would you rather satellites have both of these capabilities, or neither of these capabilities? ∘ Both scenarios should be allowed ∘ Neither scenario should be allowed

Why did you choose your answer to the previous question? [Text box] _____

[Comfort] Imagine that there were commercial satellites that could take images that capture fine-grained details, and could take images multiple times per hour. Imagine that these very high resolution satellite images do not cost any money. Consider these very high resolution satellites when you answer the bucket-sorting questions on the following pages.

Move the following location boxes based on whether you think it is appropriate for very high resolution satellites to image them. (Boxes will automatically expand as needed to fit additional entities) [Respondents must sort the following options into "Yes appropriate" and "Not appropriate"]: Mental health clinic parking lot, Grocery store parking lot, Kindergarten playground, Location of illegal business, Religious center parking lot, Rainforest, Backyards, The White House and surrounding areas, Native lands, Adult-only store parking lot

Move the following entities into two buckets for whether they should have access to very high resolution satellite imagery. (Boxes will automatically expand as needed to fit additional entities) [Respondents must sort the following options into "Yes Access" and "No Access"]: The United States government, Governments around the world, All children, Agriculture industry, Finance industry, Non-profit organizations, Convicted criminals, Romantic partners, Police, Employers

Move the following activities into whether you think it is appropriate for powerful satellites to image them. (Boxes will automatically expand as needed to fit additional entities) [Respondents must sort the following options into "Yes appropriate" and "Not appropriate"]: Driving, Building without a permit, Sunbathing, Someone using illegal drugs in their backyard, Someone using illegal drugs in public, Car accidents, People having extramarital affairs, Fishing illegally, Meeting with friends or family, Visiting a jewelry store

Choose the statement that you most agree with. Statement A: Satellite images should be available to everyone for free. Statement B: Access to satellite imagery should cost money and depend on what the user is requesting the imagery for. Statement C: The U.S. government should be the only ones allowed to access satellite imagery. Statement D: No one should have access to satellite imagery. Which statement do you agree with the most? ∘ Statement A ∘ Statement B ∘ Statement C ∘ Statement D

Why do you agree the most with the statement that you chose? If you cannot articulate a reason, please enter "Uncertain". Text box _____

Below is an image of people standing, at varying image qualities. Suppose a commercial satellite took images of you from this angle. Select the most detailed image quality you are comfortable with being publicly available. ∘ [Image of people in parking lot downsampled to

1m resolution] ∘ [Image of people in parking lot downsampled to 50cm resolution] ∘ [Image of people in parking lot downsampled to 15cm resolution] ∘ [Image of people in parking lot downsampled to 5cm resolution] ∘ [Image of people in parking lot downsampled to 1cm resolution]

Consider the most detailed image quality you chose from the previous question. What is the most frequent you would be comfortable with a satellite of that image quality taking pictures of you? ∘ Never ∘ Every year ∘ Every month ∘ Every week ∘ Every day ∘ Every hour ∘ Every minute ∘ Every second

[Attention Check] For this survey, it is important that you read the instructions carefully to ensure the results are accurate. If you are reading these instructions, please type 10 into the answer box below. [Image shown similar to Figure A.1 but this time of 8 people] _____

[Demographics] Almost done! This final page contains some demographic questions.

In which state do you currently reside? [Drop-down of all 50 states, Washington D.C., and Puerto Rico]

Which of these best describes your current gender identity? ∘ Man ∘ Woman ∘ Non-binary/third gender ∘ Other ∘ Prefer not to say

Choose one or more races that you consider yourself to be ∘ White or Caucasian ∘ Black or African American ∘ American Indian/Native American or Alaska Native ∘ Asian ∘ Native Hawaiian or Pacific Islander ∘ Other ∘ Prefer not to say

How old are you? ∘ Under 18 ∘ 18-24 years old ∘ 25-34 years old ∘ 35-44 years old ∘ 45-54 years old ∘ 55-64 years old ∘ 65+ years old ∘ Prefer not to say

What is your present religion, if any? ∘ Protestant ∘ Roman catholic ∘ Mormon ∘ Orthodox (such as Greek or Russian Orthodox) ∘ Jewish ∘ Muslim ∘ Buddhist ∘ Hindu ∘ Atheist ∘ Agnostic ∘ Something else ∘ Nothing in particular ∘ Prefer not to say

Generally speaking, do you usually think of yourself as a Republican, a Democrat, an Independent, or something else? ∘ Republican ∘ Democrat ∘ Independent ∘ Other [Text box]_____ ∘ No preference ∘ Prefer not to say

[End of Survey] [Message showing MTurk code]

Thank you for taking this survey! If you want to learn about current satellite capabilities and use cases, here is a webpage for one of the many commercial satellite companies: https://www.planet.com

A.2 Survey Codebook

These themes are based on all free response questions in the survey. The benefits and harms section of the codebook are from the questions in results Section 2.6.2. The access considerations section summarizes responses to the scenario questions (Section 2.6.3) and statement questions (Section 2.6.5). The adversaries/untrusted parties category combine free responses from all questions.

A.2.1 Potential Benefits

- Mapping
- Real Estate and Infrastructure
- Government accountability
- Legal Compliance and Crime Detection

- Targeted Advertising
- Emergency Response
- Studying Natural Phenomena
- More Information
- Surveillance
- Agriculture

A.2.2 Potential Harms

- Invasion of privacy
- Security
- Facilitating Crimes
- Military Operations
- Surveillance
- Law enforcement Abuse
- Natural Resource Misuse
- Unwanted Advertising
- Incorrect Identification
- Compliance
- AI Integrations

A.2.3 Access considerations

This list includes any factors respondents mentioned when they explained why to give/not give access to satellite imagery.

- Adversaries
- Technical Logistics
- Personal utility
- Conditional access

- Social benefit
- Privacy expectations
- Legality
- Fairness/ethics
- Innovation
- Slippery slope
- Preventing abuse
- More knowledge
- Public safety

A.2.4 Adversaries/Untrusted Parties

This includes any group respondents specifically mentioned that they didn't want to give access to satellite imagery or that they didn't trust to not misuse the satellite imagery.

- United States government
- Other world governments
- The public
- Stalkers
- Exes
- Law enforcement
- Employers
- Insurance companies

A.3 Private Activity Image Metrics Codebook

See Figure A.2 for how we grouped image metrics based on privacy concerns that people brought up. The codebook is represented in a graph (tree) format. The concerns people brought up fell under 5 high-level categories: cars, text, pets, equipment, and people (these are five child nodes under a root "entity" node in the tree). Within the people category, we

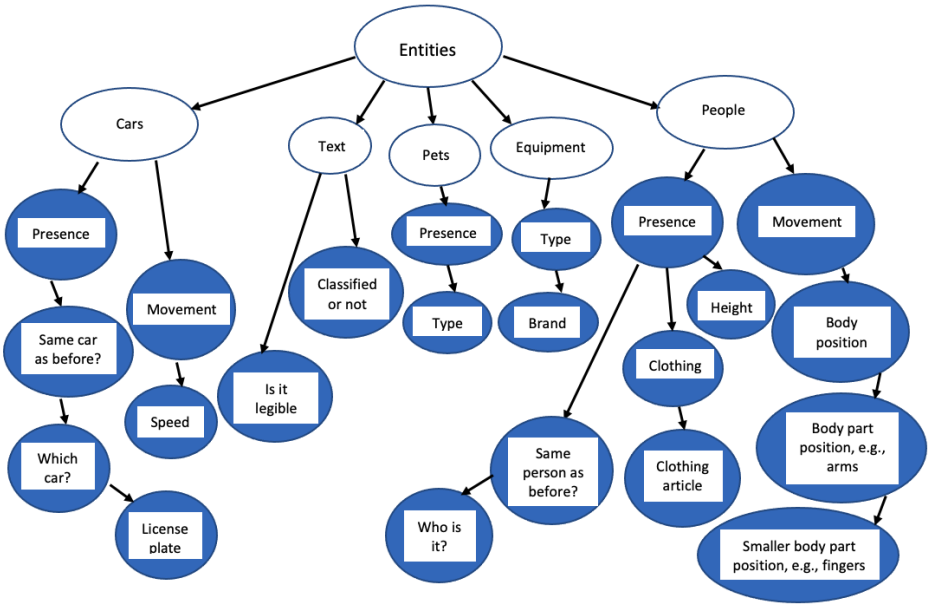


Figure A.2: Summary of private activity image metrics. Nodes that are not shaded are categories that are not themselves observable metrics. Shaded nodes are image metrics.

included a sub-category of movement (a child node). Within each category were aspects of images that would need to be observable in order to distinguish a particular activity that was deemed private by at least one person during the brainstorming session.

Each child node requires equal to or more resolution than its parent node metric. For example, knowing what clothes someone is wearing requires a better spatial resolution than knowing that a person is in the image. For another example, in the people node, the most specific movement to measure based off the brainstorming sessions is the finger position. The people movement branch of the tree could be used to predict activity type, such as running if the body position is known, and if the finger position is known, it could be used to predict as specific as finger swiping patterns on a phone.

A.4 Spatiotemporal Resolution Comfort

See Figure A.3.

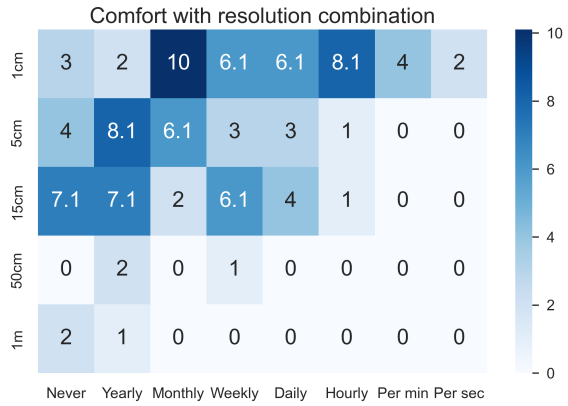


Figure A.3: Percentages of respondents who are comfortable with the given spatial and temporal resolution combinations. The most common choice was allowing 1cm spatial resolution satellite images to be taken monthly of themselves.

Appendix B

Chapter 3

B.1 Interview Protocol

Introduction Good morning/afternoon. Our names are [Researcher 1] and [Researcher 2] and we're working with [Institution 1] and [Institution 2] on this study. Thank you for coming. You are one respondent in our sample who has been asked about their university satellite club experience. We're doing this so that we can understand the challenges and complexities of building university satellites, and our ultimate goal is to have these interviews help us come up with and build future open-source tools for small satellites.

[Other introduction, recording consent.]

Experience/Background

- What satellite clubs are you a member of or have you been a member of? What years?
- What role or roles do/did you perform at the Satellite Club? For example, electrical, structures, admin, communications, etc.
- How many years of experience do you have working on satellite-related projects?
- Which satellites have you worked on developing? You can count ones that are still in progress and have not gone into space yet.

Mental Models

- For this next part, I'd like you to pick one satellite you worked on if you worked

on multiple. I'd like you to spend 3-5 minutes drawing a diagram of how all of the subcomponents of the satellite are connected together, including the ground station. I will open up a whiteboard on Zoom. If you don't want to do a Zoom whiteboard, you can draw on your preferred format instead. This doesn't have to be perfect.

- Could you walk us through the diagram?
- What goals do you have when building your satellite?
- What are the most difficult aspects of building your satellite?
- *If they haven't already talked about engineering/technical details in the previous question:* What are the most difficult technical aspects of building a satellite?
- Are there any scenarios you're concerned about when building and/or launching a satellite?

  If yes:

  - How do you address these concerns?
  - *If satellite was launched:* Did any of those scenarios end up happening?
  - *If scenarios did end up happening:* If you were to advise your past self about preparing for different scenarios, what advice would you give?

Security-Specific Questions

- *If they have not brought up security:* One type of concern we're interested in is cybersecurity concerns. From this point on, when we use the word secure or security, we mean in terms of cybersecurity. Do you or did you have any concerns related to security about your satellites? You might not have any such concerns – that's fine, and we'd like to hear about that too.
- *If they brought up security organically:* From this point on, when we use the word secure or security, we mean in terms of cybersecurity. Do you or did you have any other security concerns that you haven't mentioned yet?

- What do you think is the worst thing that could happen if the satellite is not secure?
- *If adversaries not already discussed:* From whom do you think satellites might need protection in the worst case?
- Satellite teams have many reasons to not focus on security. Was security discussed? If so, how much?
  - What were/are the conversations often about?
  - *If adversaries were not already discussed:* Did you discuss what entities your satellites might need protection from?
- How much was security discussed compared to reliability?
- What tools or techniques, if any, does or did your team already use to help with security?
- *If satellite groups have tried to make their satellites secure:* Have you experienced any challenges in trying to incorporate security into your satellite?
- *If technicaly challenges not mentioned in previous question:* What technical challenges did you have trying to incorporate security into the design or implementation stage?
- Do you think there are tools or practices your team could incorporate to help improve security for future university satellites? These tools do not need to exist yet. The answer to this question can also be no.
- How confident are you that you are able to build a satellite that is secure on a scale of 1 to 5, with 5 being the most confident? Why did you rate yourself that number?
- How would you rate your knowledge of computer security on a scale of 1 to 5, with 1 being a novice and 5 being an expert? Why did you rate yourself that number?

Demographics

- What is/was the size of your satellite team?
- What is/was the team's budget?
- What is your major / what was your major in college?
- Have you taken a security course or learned about cybersecurity? Before or after you

worked on a satellite?

- Have you worked on satellites in industry?
  - *If yes:* What differences, if any, did you notice in the security practices and tools between university and industry satellites?
  - *If there were differences:* Based on your experience, if you were to work in your satellite club again, is there anything from a security standpoint that you would do differently?

Closing Questions

- *(Ask club-specific questions)*
- Are there any other questions you expected us to ask?
- Is there anything else you want to tell us about your satellite-building experience?

B.2 Interview Response Codebook

- Goals when building satellite
  - Reliability
  - Scientific missions
  - Education of club members
  - Education/Outreach outside of club
  - Building cheaply
- Difficulties
  - Non-technical
    - Safety/harm to themselves
    - Access to resources
      - Time

        - People
        - Experience
        - Communication/Knowledge transfer
        - Money
        - Existing external documentation
        - Materials
        - Physical Testing
    - Technical
        - Specific subsystem implementation
        - Integrating subsystems together
        - Building from scratch
        - Reliability
        - Getting design stage correct
        - Minimizing complexity
        - Testing
- Security
    - Was security discussed in the interview before we explicitly asked?
        - Yes
        - No
    - Was security discussed within the club?
        - Yes
        - No
    - Respondent perceived threat likelihood
        - "Unlikely"
    - Who was concerned

    - Third-party concerns (launch provider, NASA)
    - Within club
- When did security concerns come up?
    - Security concerns discussed in club
    - Security concerns only came up when we asked in interview
- Security Issues
    - Denial of service
    - Malicious commands
    - Memory corruption
    - Remote code execution
    - Physical access pre-launch
    - Malicious third-party code
- Security Impact/Outcomes
    - Wrecking own satellite
    - Harming other satellites
    - Propelling the satellites
    - Blinding astronauts
    - Hitting International Space Station (ISS)
    - Violating regulation
    - International incident
    - Crash causing debris
- Adversary
    - “Hacker”
    - “Satellite pirates”
    - Environmental advocacy group (specific group anonymized)
    - Foreign adversary

    - Amateur radio operator
    - Personal vendetta
    - For-fun hackers (curiosity, boredom)
- Non-Adversary (when people specifically mentioned adversaries they were not concerned about)
    - Foreign adversary
- Security choices
    - No encryption
    - Protections other than encryption
    - Exploring worst-case scenarios
    - Encrypting uplink
    - Obfuscation
        - Of code (not public)
        - Of commands
- Security Tradeoffs
    - Intended public downlink
    - Open-source implementation
    - Maintaining simplicity
    - Maintaining functionality
- Challenges Incorporating Security
    - None/not sure
    - Misunderstanding of regulation (e.g., thinking you cannot encrypt amateur radio telecommands to satellites)
    - Monitoring code across systems/people
    - Time/effort

    - Expertise
- Security Tools Used
    - None/not sure
    - Fuzzers
    - Manual auditing
    - "Hardware debugging"
    - Encryption
    - "Frequency modulation"
    - Obfuscation
- Industry Security Practice Differences
    - Equal practices between university and industry
    - Industry more secure
    - Lack of good security practices in industry
    - No satellite industry experience
- Future Security Ideas
    - None/not sure
    - Non-technical
        - Check organizational security (e.g., document access)
    - Technical
        - Check security of third-party software libraries
        - Vulnerability testing
        - "Public key/private key hashing"
        - Fuzzing tools
        - Encryption tools
        - More secure versions of existing tools (e.g., OpenLST)

- Documentation
  - Example code
  - Best practices descriptions

B.3 Repository Information

B.3.1 Repository Statistics per Satellite Club

Tables B.1, B.2, and B.3 show repository statistics for each satellite club where we performed a code analysis. For each group, we manually categorize repositories based on both the subsystem they are for and the type of design collateral they contain. We collect this information by first cloning each repository from the satellite clubs' GitHub organizations. Then, we manually categorize each repository and parse the Git histories to extract author/commit information. Note that totals may not sum up across the rows, since repositories may address multiple subsystems or design collateral types.

Entries marked with asterisks (*) represent outliers which we can explain in the data:

- Table B.1 (miscellaneous commits): this satellite club used an automated script to update some of their repositories, hence the high commit count.
- Table B.3 (flight computer authors): this satellite club forked an industry-standard boot firmware into their organization. Our Git author counts also include all (non-satellite club) committers who contributed to the fork's upstream repository.

B.3.2 Number of Files per Software Type

We now focus on only the repositories we categorized as containing "software". Tables B.4, B.5, and B.6 show the number of source code files for each of the satellite club's most commonly used languages. This data was collected by using the commonly used `cloc` [1]

[1] `https://linux.die.net/man/1/cloc`

Satellite A	Hardware			Software			Infrastructure			Documentation			**Total**		
	R	A	C	R	A	C	R	A	C	R	A	C	R	A	C
Ground Station	0	0	0	11	35	786	0	0	0	2	7	280	13	35	1066
Sensors	0	0	0	5	10	63	1	2	4	0	0	0	6	10	67
Communications	1	3	24	5	13	217	1	1	8	0	0	0	7	16	249
Bus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Power	3	3	9	0	0	0	0	0	0	0	0	0	3	3	9
Flight Computer	1	14	256	9	38	1145	1	1	2	0	0	0	11	51	1403
Actuators	0	0	0	1	4	39	0	0	0	0	0	0	1	4	39
Educational	0	0	0	3	9	62	0	0	0	2	3	67	5	10	129
Website	0	0	0	8	32	330	2	9	34	2	6	24	12	39	388
Miscellaneous	0	0	0	4	146	6339*	3	115	5782*	0	0	0	7	146	12121
Total	5	19	289	40	244	8732	8	125	5830	6	15	371	53	270	9135

Table B.1: Number of **R**epositories, unique **A**uthors, and **C**ommits per subsystem and type for Satellite A.

utility, which counts the number of source lines for a large array of recognizable programming languages.

For all satellite clubs, but especially Satellite A, we see a high number of C/C++ headers. This is because these satellite clubs vendored various SDKs for embedded devices directly into their repositories. Satellite A also implemented a mobile app, the repository for which includes vendored Objective-C libraries.

B.3.3 Number of Code Lines per Software Type

Tables B.7, B.8, and B.9 use the same underlying data from `cloc`, but instead display the number of lines characterized as "source code" (as opposed to "blank lines" or "comments"). Again, C/C++ headers from vendored SDKs comprise a large proportion of the overall

Satellite B	Hardware			Software			Infrastructure			Documentation			**Total**		
	R	A	C	R	A	C	R	A	C	R	A	C	R	A	C
Ground Station	0	0	0	1	13	462	0	0	0	0	0	0	1	13	462
Sensors	4	4	62	7	32	521	1	1	31	0	0	0	12	35	614
Communications	1	1	11	2	5	27	0	0	0	0	0	0	3	5	38
Bus	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Power	5	6	79	2	5	45	1	1	11	0	0	0	8	7	135
Flight Computer	3	6	54	5	40	2199	1	1	3	1	37	2138	10	44	4394
Actuators	8	11	75	7	10	131	4	10	62	0	0	0	19	18	268
Educational	1	8	12	3	10	49	0	0	0	3	7	20	7	20	81
Website	20	0	0	0	0	0	2	2	9035	1	7	78	3	9	9113
Miscellaneous	3	8	62	4	22	546	2	12	63	0	0	0	9	29	671
Total	24	22	328	30	88	3518	11	22	9205	5	47	2236	60	107	12916

Table B.2: Number of **R**epositories, unique **A**uthors, and **C**ommits per subsystem and type for Satellite B.

codebase. Satellite C also includes two huge Rust files which appear to be automatically generated libraries for interfacing with specific microcontrollers.

Satellite C	Hardware			Software			Infrastructure			Documentation			**Total**		
	R	A	C	R	A	C	R	A	C	R	A	C	R	A	C
Ground Station	1	19	282	2	19	292	0	0	0	0	0	0	3	19	574
Sensors	6	43	920	16	73	3285	3	22	1848	2	4	10	27	99	6063
Communications	5	35	1266	10	50	1255	1	2	2	0	0	0	16	72	2523
Bus	9	71	2443	3	15	573	6	49	668	1	2	7	19	83	3691
Power	5	30	817	2	9	167	4	17	355	0	0	0	11	30	1339
Flight Computer	4	36	930	10	3444*	97028*	5	31	1986	1	2	7	20	3463	99951
Actuators	2	11	109	1	17	1757	1	17	1757	1	2	3	5	29	3626
Educational	0	0	0	0	0	0	0	0	0	5	26	175	5	26	175
Website	0	0	0	4	14	128	1	2	10	1	2	5	6	15	143
Miscellaneous	2	33	617	6	19	368	4	13	317	1	1	2	13	52	1304
Total	32	157	7172	50	3571	102745	22	100	5172	9	29	192	90	3664	109584

Table B.3: Number of **R**epositories, unique **A**uthors, and **C**ommits per subsystem and type for Satellite C.

Satellite A	C	C++	C/C++ Header*	Java	Javascript	Objective-C*	Python	**Total**
Ground Station	16	230	13472	148	120	884	610	15480
Sensors	60	0	264	0	0	0	6	330
Communications	0	0	0	0	131	0	24	155
Bus	0	0	0	0	0	0	0	0
Power	0	0	0	0	0	0	0	0
Flight Computer	268	0	1115	0	0	0	4	1387
Actuators	0	0	0	0	0	6	0	6
Educational	35	0	199	0	0	0	0	234
Website	3	211	13427	148	547	884	0	15220
Miscellaneous	52	32	250	0	77	0	0	411
Total	330	262	14786	148	757	884	644	17811

Table B.4: Number of source code files per subsystem and language for Satellite A.

Satellite B	Arduino Sketch	C	C++	C/C++ Header	Python	Ruby	**Total**
Ground Station	0	0	0	0	8	88	96
Sensors	3	27	49	113	20	24	236
Communications	0	0	0	0	35	0	35
Bus	0	0	0	0	0	0	0
Power	6	0	0	0	0	0	6
Flight Computer	6	216	16	275	21	216	750
Actuators	1	5	11	21	3	0	41
Educational	0	4	7	8	0	0	19
Website	0	0	0	0	0	0	0
Miscellaneous	2	2	5	14	8	88	119
Total	18	254	88	431	87	275	1153

Table B.5: Number of source code files per subsystem and language for Satellite B.

Satellite C	Assembly	C	C++	C/C++ Header	Python	Rust	**Total**
Ground Station	0	22	0	300	2	0	324
Sensors	274	554	20	1076	110	2*	2036
Communications	0	130	3	395	22	18	568
Bus	0	25	0	18	68	0	111
Power	0	0	3	4	10	0	17
Flight Computer	0	48	0	45	108	3	204
Actuators	0	132	2	176	5	0	315
Educational	0	0	0	0	0	0	0
Website	0	0	0	0	0	0	0
Miscellaneous	1	1	0	0	36	0	38
Total	275	728	26	1217	311	23	2580

Table B.6: Number of source code files per subsystem and language for Satellite C.

Satellite A	C	C++	C/C++ Header	Java	Javascript	Objective-C	Python	**Total**
Ground Station	5850	49144	1703803	33883	7455	113530	142245	2055910
Sensors	12532	0	128926	0	0	0	771	142229
Communications	0	0	0	0	6014	0	1857	7871
Bus	0	0	0	0	0	0	0	0
Power	0	0	0	0	0	0	0	0
Flight Computer	59954	0	448973	0	0	0	245	509172
Actuators	0	0	0	0	0	771	0	771
Educational	3630	0	108735	0	0	0	0	112365
Website	4773	47687	1702967	33883	110853	113530	0	2013693
Miscellaneous	4539	6591	114756	0	151790	0	0	277676
Total	71825	55735	2209085	33883	268059	113530	145118	2897235

Table B.7: Number of source code lines per subsystem and language for Satellite A.

Satellite B	Arduino Sketch	C	C++	C/C++ Header	Python	Ruby	**Total**
Ground Station	0	0	0	0	1645	1776	3421
Sensors	203	15640	8045	60076	1069	264	85297
Communications	0	0	0	0	3515	0	3515
Bus	0	0	0	0	0	0	0
Power	325	0	0	0	0	0	325
Flight Computer	700	45408	10893	44224	1224	11888	114337
Actuators	60	1142	568	971	114	0	2855
Educational	0	209	174	203	0	0	586
Website	0	0	0	0	0	0	0
Miscellaneous	63	274	114	14429	1645	1776	18301
Total	1351	62673	19794	119903	7567	13522	224810

Table B.8: Number of source code lines per subsystem and language for Satellite B.

Satellite C	Assembly	C	C++	C/C++ Header	Python	Rust	**Total**
Ground Station	0	6018	0	72450	80	0	78548
Sensors	85802	234261	3633	1059330	8804	449665*	1841495
Communications	0	37138	268	77353	1120	8145	124024
Bus	0	4016	0	442	5113	0	9571
Power	0	0	515	238	1146	0	1899
Flight Computer	0	5741	0	7338	10239	364	23682
Actuators	0	26041	43	45936	509	0	72529
Educational	0	0	0	0	0	0	0
Website	0	0	0	0	0	0	0
Miscellaneous	140	6	0	0	3591	0	3737
Total	85942	274719	4416	1068598	26935	458174	1918784

Table B.9: Number of source code lines per subsystem and language for Satellite C.

Appendix C

Chapter 4

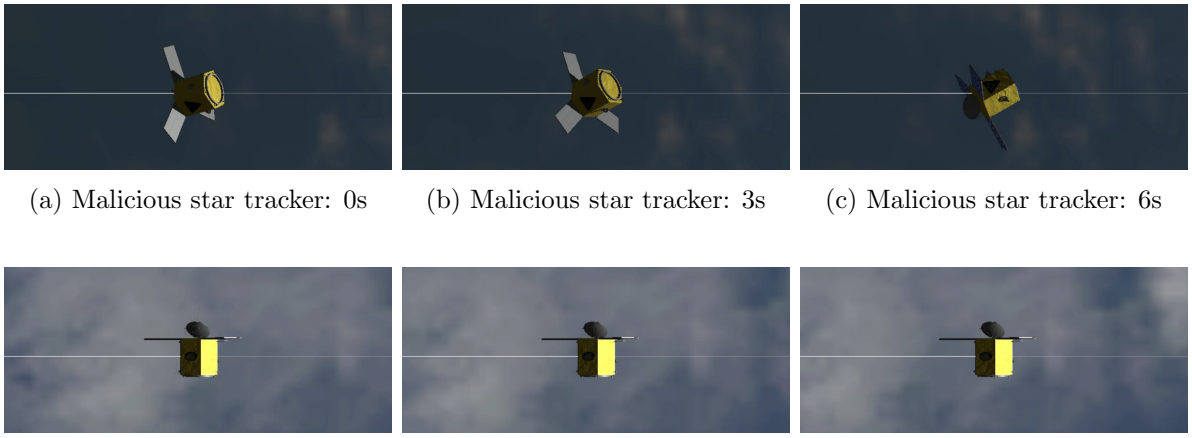
C.1 Attack Visualization

C.1.1 Example Attack Visualizations

In order to conceptualize the attacks, in this section, we provide a set of visualizations of the impact of the attacks. Figure C.1 shows a visualization of the simulation under a manipulated data attack, as well as the honest scenario, without any defenses. The attack clearly causes the satellite to tumble uncontrollably, whereas with an honest star tracker, the satellite can always orient itself properly relative to Earth.

Additionally, in Figure C.2, we show the difference between the current and target quaternion values over time. In particular, Figure C.2g and Figure C.2h show an example of the manipulated data attack when the satellite uses multiple star trackers or only one star tracker, respectively. However, in either case, the quaternions never converge to the goal orientation.

In the case of a burst size of 3 with no defenses in place, interesting behavior is visible through time series plots. Figure C.2b and Figure C.2e show the difference across time between the current quaternion (orientation) values for the satellite and the target quaternion values, which can be seen as an error metric simpler than the one we describe above. With redundancy, there is slight performance degradation, as convergence happens longer compared to nominal timing conditions. When there is no redundancy, the satellite achieves "marginal stability", meaning it is not completely unstable or completely stable. This could still cause negative effects, such as requiring the reaction wheels to do extra physical work to attempt to achieve convergence, or failure of the mission entirely due to the inability to



(a) Malicious star tracker: 0s (b) Malicious star tracker: 3s (c) Malicious star tracker: 6s

(d) Honest star tracker: 0s (e) Honest star tracker: 3s (f) Honest star tracker: 6s

Figure C.1: A malicious star tracker can cause a satellite to tumble using the Data Manipulation attack (top), whereas, with an honest star tracker, the satellite would maintain its attitude (bottom).

orient properly.

C.2 Kalman Filter

Below we describe in detail the components in our satellite attitude determination extended Kalman filter. This Kalman filter will be used to enhance the realism of our simulation since many complex satellites use them in their guidance, navigation, and control systems (GNC). The Kalman filter will take as inputs each relevant ADCS sensor that we simulate, specifically three star trackers and one IMU (a set of three-axis gyroscopes) to the mission, all of which input into an attitude determination Kalman filter similar to what might exist on a real mission for a government or commercial satellite. Gyros are commonly used in addition to attitude sensors [96].

C.3 What is a Kalman Filter?

A Kalman filter is used to generate an estimate of the true state of a control system by combining sensor input, a mathematical model of system dynamics, and hardware-specific parameters.

The main two steps in every Kalman filter are the prediction and update steps. Before receiving a measurement of the attitude state from star trackers, the prediction step calculates what its guess as to the next state will be. Then, the update step takes into account the sensor measurement. The amount to trust the sensor measurement versus the state estimated in the prediction step is determined by a variable called the *Kalman Gain*. This gain amount is calculated dynamically at each timestep, as the Kalman filter is designed to be able to trust the prediction versus measurements can differently across time.

At each timestep, the Kalman filter keeps track of two state vectors. The first is a vector of the attitude and angular rate estimate. The second is the error state, i.e., how much we should trust our estimate.

C.4 Extended Kalman Filters

The most common form of state estimate for satellite attitude sensors is an extended Kalman filter. In linear systems, the traditional Kalman filter can be used. But because attitude dynamics are a nonlinear system, the Kalman filter linearizes the physics estimation around the current timestamp.

More specifically, the most common EKF for satellite attitude determination is the *multiplicative* extended Kalman filter (MEKF). This means that unlike additive filters, the quaternion uses a multiplicative update to update its state.

C.5 EKF Components for ADCS

Below we show how an EKF works for an attitude determination algorithm, using three star trackers and one IMU (three-axis gyro).

C.5.1 Prediction step

At each timestamp t, the prediction step predicts what the next state will be *before* the star tracker measurements are even considered.

Covariance prediction:
$P_{k+1}^{-} = \Phi P_k^{+} \Phi^T + C\Delta t$

State prediction:
$x_{k+1}^{-} = f(q_k^{+}, \omega_{k+1})$

Below are the variables and equations involved in predicting the covariance and state. Quaternions are predicted based on a linearized, discretized mathematical model. The gyro bias state prediction is assumed to be constant, so bias only changes in the correction step of the Kalman filter.

The State Equation

Attitude is not the only part of the state we're dynamically estimating. We will use gyroscopes to help with estimating the attitude through their angular rate measurements. The full state vector x_k will be 7x1, where the first four components are the quaternions (attitude state) and then the last three components are gyro drift-rate bias in the x, y, and z directions. Note that for our evaluation, we did not include bias estimation for proof-of-concept tests in ControlPhoenix, but we plan to add this IMU and bias noise back in in future work. Therefore, the three bias terms here are held to 0's.

$$x_k = \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \\ b_x \\ b_y \\ b_z \end{bmatrix}$$

Attitude State

Attitude state vector. The attitude state q_k is a 4 x 1 vector representing 4 attitude quaternions. This is a subcomponent of the whole 7x1 state equation, which also includes the 3x1 set of biases, which tracks a bias for each of the three gyros.

Attitude state prediction. In the first step, the quaternion at time $k+1$ is predicted by integrating the angular rate, and adding it to the formerly computed quaternion. Because there is no close-formed solution to this, an approximation is instead required. This is like taking a small step in the direction of the derivative of the most recently measured angular rate.

The time derivative of the quaternion would be $\frac{d}{dt}q_{k+1}^{-} = \frac{1}{2}\Omega(\omega_{k+1})q_k^{+}$, where

$$\Omega = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}$$

After discretizing and linearizing, we get the following matrix for how much the quaternion

is expected to rotate [11]:

$$\mathbf{F} = \begin{bmatrix} 1 & -\frac{\Delta}{2}\omega_x & -\frac{\Delta}{2}\omega_y & -\frac{\Delta}{2}\omega_z \\ \frac{\Delta}{2}\omega_x & 1 & \frac{\Delta}{2}\omega_z & -\frac{\Delta}{2}\omega_y \\ \frac{\Delta}{2}\omega_y & -\frac{\Delta}{2}\omega_z & 1 & \frac{\Delta}{2}\omega_x \\ \frac{\Delta}{2}\omega_z & \frac{\Delta}{2}\omega_y & -\frac{\Delta}{2}\omega_x & 1 \end{bmatrix}$$

The ω components are from the gyroscope readings.

Putting this together, the quaternion state is predicted as:
$q_{k+1}^{-} = \mathbf{F}_k q_k^{+}$

Error and Covariance

The Kalman filter keeps tracks of error estimates in the state prediction, which affects the state that it outputs. The error state is the estimate of the current error in the state, which is used in the update step to correct the state after the state has been predicted and the measurement has been received. The covariance matrix represents the level of uncertainty in the system. The covariance matrix changes over time. It usually grows or changes shape during prediction, and shrinks during measurement update.

State-error vector. The error state can be represented by a 6 x 1 vector. It represents an estimate of how far off the filter's current estimate is from the true state:

$$\delta x_k = \begin{bmatrix} \delta\Theta_x \\ \delta\Theta_y \\ \delta\Theta_z \\ \delta b_0 \\ \delta b_1 \\ \delta b_2 \end{bmatrix}$$

Initialization: This state-error vector is only used in the update step, as shown below. It is re-initialized after each Kalman filter update step back to 0's.

Error Covariance Matrix. This matrix P 6x6 represents how uncertain the error state is. It describes uncertainties of each component, as well as how the components are correlated. [96]

$$P = \begin{bmatrix} \sigma_{\theta,0}^2 I_3 & 0 \\ 0 & \sigma_{b,0}^2 I_3 \end{bmatrix}$$

Initialization: Initialization can be a diagonal matrix, where the first three components are initial attitude-error variances, in rad^2. The last three are initial gyro-bias variances, in $(\text{rad/s})^2$. Because we are assuming there is no information about the state during initialization, we should initialize the covariance to be very high.

Error State Transition Matrix. A 6x6 matrix describing error transition

$$\Theta = \begin{bmatrix} I_3 - [\omega_k]_x \Delta t & -I_3 \Delta t \\ 0 & I_3 \end{bmatrix}, \text{ where}$$

$$[w_k]x = \begin{bmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{bmatrix}$$

Process Noise Matrix. The process noise matrix C 6x6 is uncertainty added to the current timestep. The diagonal components correspond with gyro white-noise intensity and bias random walk. The top right and bottom left are cross-correlation between the two. We can set these to the following:

$$C = \begin{bmatrix} \sigma_g^2 I_3 & 0 \\ 0 & \sigma_b^2 I_3 \end{bmatrix}$$

Since this is the prediction step before measurements are received, star tracker noise is not considered here. It is considered later in the covariance update step in the form of the measurement noise matrix R.

Error State Covariance Prediction. $P_{k+1}^{-} = \Phi P_k^{+} \Phi^T + C\Delta t$

where Φ is the error state transition matrix above, and C is the process noise.

C.5.2 Update Step

The update step updates the attitude state, attitude error state, and the error-state covariance.

Measurement noise covariance R. A discrete Gaussian white noise process based on sensor-specific capabilities.

Measurement sensitivity matrix H. Represents what state errors impact the difference between the actual quaternion measurement and the predicted quaternion. This difference

is directly sensitive to the attitude error and not bias error:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Kalman Gain. The Kalman gain is a variable used to determine how much to trust the prediction versus the measurement in the update step, and is recomputed at each timestep. The Kalman gain is calculated as:

$$K_k = P_k^- H_k^\top (H_k P_k^- H_k^\top + R)^{-1}$$

Intuitively, the gain variable (trust amount) goes up with less sensor noise, less process noise, and lower error covariance.

The dimensions are 6 x 3, because there are 6 components where the attitude is represented in 3 components and then the gyro bias error is 3 components.

Updating Covariance. Updating covariance relies on the Kalman gain and measurement sensitivity matrix H:

$$P_{k+1}^+ = (I - K_k H_k) P_k^-$$

Intuitively, the uncertainty in the state estimate decreases directly after a measurement.

Updating Nominal State

Receiving Measurement. The measurement from a star tracker, z_{k+1} is a 4x1 vector in the form of quaternions.

Innovation. The innovation is the difference between the actual quaternion measurement z^q_{k+1} and the predicted state. First, the difference is calculated on the two 4 x 1 vectors:

$$\delta q_{\text{meas}} = z^q_{k+1} \otimes q^-_{k+1},$$

where the $\otimes$ symbol means a special quaternion multiplication.

Then, this 4 x 1 relative quaternion is converted into a 3 x 1 innovation y_{k+1} to be used in the MEKF:

$$y_{k+1} = 2\text{vec}(\delta q_{\text{meas}}).$$

The vec function is used to convert the 4 x 1 quaternions to a 3 x 1 attitude representation. It simply takes the last three components of the quaternion, chopping off q_0.

Error-state Correction. The error-state correction amount is the innovation weighted by the Kalman gain. A higher Kalman gain means it trusts the measurements more, which is why the correction is higher if the gain is higher.

$$\delta x^+_k = K_{k+1} y_{k+1}$$

Dimensions: 6 x 3 gain times a 3 x 1 innovation is 6 x 1 correction, because this correction is done for both attitude and gyro bias parameters.

The first 3 parameters of the 6 x 1 error state are attitude. We want to convert this 3 x 1 attitude representation back to quaternions to update the quaternion state:

$$\delta q_k^+ \approx \begin{bmatrix} 1 \\ \frac{1}{2}\delta\Theta^+ \end{bmatrix}$$

δq_k^+ is now a 4x1 vector.

What about the bias error state correction? The bias error state correction is also estimated with the above equation despite only multiplying the Kalman gain with the quaternion innovation and not any sort of gyro bias innovation. At a high level, this is because a high innovation (difference between measurement and estimation) is inherently dependent on the accuracy of the gyro states, and the accuracy of the gyro states are dependent on the accuracy of the bias estimates. Thus, a higher quaternion innovation implies a higher error in the gyro bias component.

Final Update Step. Finally, update the nominal quaternion by moving it by the error correction amount:

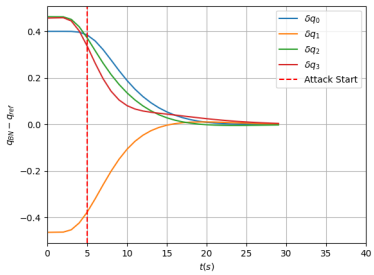
$$q_{k+1}^+ = \delta q_{k+1} \otimes q_{k+1}^-$$

The gyro bias update is calculated as:

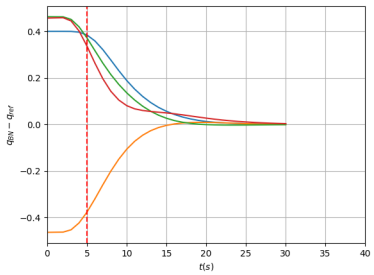
$$b_{k+1}^+ = b_{k+1}^- + \delta b_{k+1}$$

Handling Redundant Sensors. Because there are multiple star trackers, we must decide on a sensor fusion technique to take all of the star tracker messages into the EKF. Our approach, in the case where the star trackers are all sending data at very similar times (which is the case for our non-attack scenarios), is pretty similar to the sequential filtering approach [164]. At every ADCS poll period, it drains all messages from the queue. Within that poll period, the EKF will do one singular prediction step and then number of updates steps equivalent to however many star trackers were received in that period. E.g., star tracker 1 message

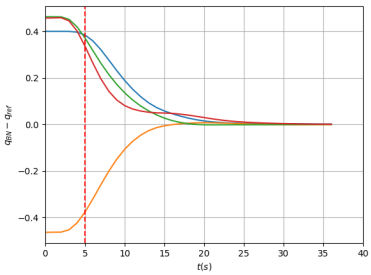
is received, which creates a new quaternion estimate after updated, $q^+_{1,k+1}$ based on the quaternion in the prediction step, q^-_{k+1}. Then, $q^+_{1,k+1}$ is input into the next update step, along with star tracker 2 message received, to output $q^+_{2,k+1}$, and so on. The timestamps for all star trackers received in the same poll period are treated as equal.



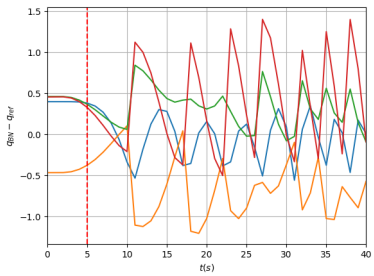
(a) Honest. δ_{max}=0.1. No redundancy



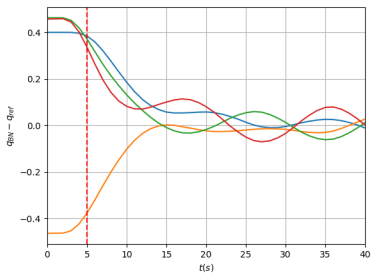
(b) Burst size 3. With redundancy. No LC thresholds.



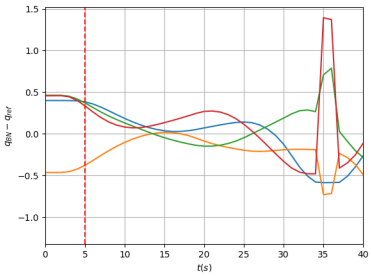
(c) Burst size 6. With redundancy. No LC thresholds



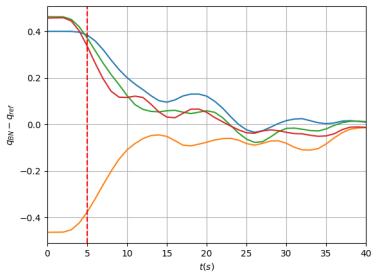
(d) Honest. δ_{max}=0.0001. No redundancy



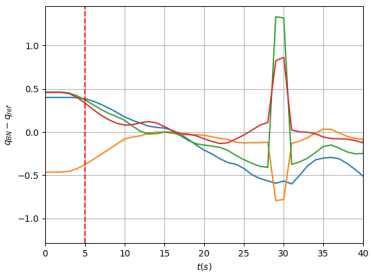
(e) Burst size 3. No redundancy. No LC thresholds.



(f) Burst size 6. No redundancy. No LC thresholds.



(g) Data Manipulation. δ_{max}=0.1. Redundancy



(h) Data Manipulation. δ_{max}=0.1. No redundancy.

Figure C.2: Time versus attitude error of different cases of attacks. The four lines are the four quaternion values making up the satellite's orientation. The attitude error for each quaternion is the current quaternion value subtracted by the target quaternion value. The red dotted vertical line marks when the attack starts. If the quaternion lines end before the x-axis limit, this means that it passed at that point and exited the test.