The Final Security Frontier: Using Privacy-Preserving Computation to Secure Satellite Rendezvous and Proximity Operations

Caroline M. Brandon*
L3Harris Technologies, Inc.
Melbourne, Florida, USA
caroline.brandon@l3harris.com

Carson Stillman*
University of Florida
Gainesville, Florida, USA
carson.stillman@ufl.edu

Joel Hirschmann
University of Florida
Gainesville, Florida, USA
joelhirschmann@ufl.edu

Sara Rampazzi
University of Florida
Gainesville, Florida, USA
srampazzi@ufl.edu

Marina Blanton
University at Buffalo
Buffalo, New York, USA
mblanton@buffalo.edu

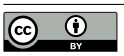
Christopher D. Petersen
University of Florida
Gainesville, Florida, USA
c.petersen1@ufl.edu

Kevin R. B. Butler
University of Florida
Gainesville, Florida, USA
butler@ufl.edu

Abstract

Space is emerging as a critical domain for secure and privacy-preserving computing, driven by the rapid growth of commercial satellites and the increasing complexity of in-space operations. This need is particularly evident in satellite rendezvous and proximity operations (RPO), where multiple satellites must compute with private, identifying, or even classified information in order to operate safely at close distances. Secure multiparty computation (MPC) offers a promising foundation for privacy-preserving collaboration, but its suitability for in-space applications remains largely unexplored. This work provides a domain-informed analysis of satellite RPO and private computation techniques for two fundamental scenarios: collision avoidance and multi-point inspection. We design and evaluate a secure two-party computation for collision avoidance and a three-party computation for multi-point inspection using the MP-SDPZ compiler. Testing on radiation-tolerant NVIDIA Jetson hardware, we find that collision avoidance can be performed securely in 7.38 seconds in a semi-honest dishonest majority model, while multi-point inspection can be performed securely in 0.28 seconds in the semi-honest model and in 2.7 seconds in the malicious model. These findings demonstrate that MPC is both feasible and practical for privacy-preserving RPO under realistic hardware and threat assumptions. To our knowledge, this is the first work to provide granular assessment and experimentally validated MPC for in-space cooperative operations, opening a new direction for cybersecurity research in emerging space systems.

*Both authors contributed equally to this research. Work primarily performed when lead authors were at the University of Florida.



WiSec '26, June 30-July 03, 2026, Saarbrücken, Germany

ACM ISBN 979-8-4007-2201-1/2026/06
https://doi.org/10.1145/3765613.3797447

CCS Concepts

• **Security and privacy → Domain-specific security and privacy architectures**; *Systems security*; *Cryptography*.

Keywords

Privacy preserving computation, secure multiparty computation, cryptography, satellites, space system privacy, rendezvous and proximity operations

ACM Reference Format:
Caroline M. Brandon, Carson Stillman, Joel Hirschmann, Sara Rampazzi, Marina Blanton, Christopher D. Petersen, and Kevin R. B. Butler. 2025. The Final Security Frontier: Using Privacy-Preserving Computation to Secure Satellite Rendezvous and Proximity Operations. In *Proceedings of the 19th ACM Conference on Security and Privacy in Wireless and Mobile Networks (WiSec '26), June 30-July 03, 2026, Saarbrücken, Germany*. ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3765613.3797447

1 Introduction

Space activities are essential to modern society for navigation, communication, and critical infrastructure for civil, economic, and national security services. As satellite numbers grow rapidly, exceeding 11,000 actively in orbit as of March 2025 [59] and projected to surpass 60,000 within the decade [68], autonomous coordination directly between spacecraft, not solely through ground stations, is becoming increasingly necessary [95]. These capabilities are required for **Rendezvous and Proximity Operations** (RPO), a class of control activities that include docking and other close-proximity maneuvers, and require precise and direct inter-satellite communication [9]. RPO also enables emerging capabilities for in-space servicing, assembly, and manufacturing operations (ISAM), such as repair and refueling, that extend the lifetime of the satellite and reduce costs [62]. However, as the number of government, commercial, and academic stakeholders increases, space becomes both more congested and contested [25]. Full trust cannot be assumed among all participants in all coordinated activities. Certain sensitive data

values exchanged during coordination, such as ephemerides (position and trajectory data) for collision avoidance or other on-board metrics, can inadvertently expose proprietary satellite designs or mission details as they are inherent properties of the satellite's sensors and design. This risk is known as **satellite characterization** [76, 83], where sharing operational data allows other parties to infer proprietary or strategically important information about a spacecraft or its mission. Specifically, sharing sensor covariance matrices during satellite coordination can expose sensor specifics and mission trajectories, even between allies [83]. Current RPO algorithms require cleartext sharing of covariance data, creating an unaddressed privacy vulnerability in autonomous space operations. Thus, protecting privacy while ensuring safe and cooperative operations is a fundamental challenge for the future of autonomous space systems. This work aims to protect covariance matrices using **secure multiparty computation (MPC)**. The main contributions of this paper are:

- **Identifying the Need for Computational Privacy in RPO:** We expose a previously-overlooked vulnerability in modern satellite architectures: the assumption of full trust in shared covariance and state-estimation data. We show how MPC enables privacy-preserving coordination between satellites operating in close proximity while maintaining physical safety.
- **Testing Privacy-Preserving RPO Algorithms Across Adversarial Models:** We categorize representative mission scenarios according to realistic adversarial assumptions, ranging from cooperative but curious partners to actively malicious participants. This mapping bridges mission security needs with established cryptographic threat models. We design privacy-preserving variants of state-of-the-art RPO algorithms using the MP-SPDZ [40] compiler, based on understanding which inputs require protection.
- **Characterizing MPC Performance Under Realistic Space Constraints:** We evaluate MPC performance under realistic orbital communication limits and on radiation-tolerant hardware, demonstrating that two- and three-party MPC can meet operational timing and safety requirements in many RPO scenarios. Two-party collision avoidance executes securely in 7.38 seconds and multi-point inspection in 0.28 seconds (semi-honest) and 2.7 seconds (malicious model).

To our knowledge, we are the first to provide a granular evaluation of different MPC protocols, to test on space-hardened hardware, and to consider satellite multi-point inspection. This work addresses a relevant research gap regarding the criticality of space systems and the increasing number of collaborative operations in space. We demonstrate that with certain mission-specific parameters, these operations are currently feasible using MPC but that deployments must be performed with careful consideration of the adversarial model and space constraints. Our code is available on Zenodo.[1]

2 Background

Cyberattacks related to the space domain have intensified in recent years, highlighting vulnerabilities with space assets and motivating proactive security solutions [50, 71]. Security is especially needed as the number of stakeholders increases, particularly within the competitive commercial space sector, where most technological innovation and cost reductions occur [53]. The US Space Force Strategy guides highlight the importance of both joint coalitions between commercial, inter-agency, and international partners, and the need to maintain competitive advantage [30, 67]. Protecting mission specifications and proprietary design information is critical to achieving this goal, but a great deal of information must be shared between stakeholders in space to preserve physical safety and operational success [67]. Therefore, there is a problem of satellites sharing certain data (e.g. ephemeris covariance information or state-of-health telemetry) that can leak proprietary design and potentially mission goals.

Rendezvous and Proximity Operations. Satellite operations often occur when satellites are at large distances from each other. Even in low Earth orbit (LEO), satellites typically operate on the order of megameters apart [69], and have days or weeks to coordinate their trajectories. Rendezvous and proximity operations (RPO) is a class of satellite operations that occur at much closer distances and conduct on-board trajectory adjustments in near-real time [74]. RPO, which is housed in the guidance navigation and control (GNC) section of the satellite bus, is necessary for multi-agent space systems when the distances between satellites are 500 km or less and are largely autonomous [74, 79]. This is critical in both co-orbital and constellation satellite configurations, enabling operations such as collision avoidance and in-space repair.

Rendezvous indicates that two or more satellites are in the same plane, attitude, and phasing, and *proximity operations* refer to tasks in which two or more satellites in approximately the same orbit perform intentional maneuvers with each other, affecting their relative states [79]. These activities include on-orbit servicing (OSS), docking, refueling, formation flying, and debris removal, all operations that require precise control, using tools such as the global navigation satellite system (GNSS), radar, LiDAR, or optical sensors [63]. RPO encompasses many other space specialties, such as sensor advancement, data processing and sharing techniques, orbital dynamics, and more, and has been used in a variety of missions, including the international space station (ISS) and mission extension vehicles (MEVs), which extend the lifetime of other spacecraft [9, 35]. It is most relevant to satellites in dense orbits, specifically LEO, where about ninety percent of satellites operate. RPO is mainly used in small satellites, a mid-range class of spacecraft.

A motivating factor in RPO development is the growing need for autonomy. RPO maneuvers are time-critical due to the close ranges of the satellites involved, and depend on readily available computation and decision-making data [74]. The reliability of ground stations can be poor and difficult to predict. For LEO, the ground station has at best a 12 to 20 minute window of contact within a typical satellite's 90 minute orbit [13]. In the worst case, it could take up to a day to establish communication with the satellite [92]. Thus, satellites rely on crosslinks and autonomous in-space capabilities, such as handoffs between internet satellites, for reliable coverage. Additional details on satellite communications are given in Appendix A.

Covariance Matrices. In current RPO calculations, satellites will share some data in cleartext, including satellite ephemerides and

[1] https://doi.org/10.5281/zenodo.18565977

mission data, as well as characterization information, such as satellite dimensions or radio-frequency information [4]. The exact quantification of uncertainty is represented by the covariance matrices associated with these data points, typically using a covariance-adaptive Kalman Filter, an approach for tracking and data prediction tasks [74, 96]. Performing RPO calculations safely requires that these matrices be communicated as inputs in joint computations. However, prior research in other cyber-physical systems has demonstrated that covariance matrices can be used to characterize the sensors or instruments used and thereby leak sensitive information about the satellite's design or mission [65, 76, 83]. Our solution involves using privacy-preserving computation to protect these covariance matrices while enabling joint computation on the required data.

3 Related Works

Research related to securing the space segment and on-board systems is limited but growing. Existing work has examined the cyberphysical security of satellites [21, 94], communications and networking security [46, 58], hardware and embedded systems [37], security-by-design [47], and artificial intelligence [91]. Others have classified threats to satellite security, addressing some outdated assumptions regarding space security and vulnerabilities present in the software of current assets [28, 87, 93]. These do not consider autonomous security and privacy measures in space. Several works have addressed the issue of satellite characterization as a known issue, particularly when covariance matrices are known, and the subsequent risk of unauthorized inference of operations and design [10, 29, 76, 83]. Additional research has examined the use of secure data computation for space applications, including a decentralized federated data ecosystem as a trusted third party for secure computation [33], protecting satellite imaging data during on-board processing [78], and addressing privacy concerns of two-way ranging satellite broadcast systems [18].

Others looked into securing satellite conjunction analysis with MPC or homomorphic encryption [34, 38, 49]. Although these works offer useful proofs-of-concept for secure computation in space and demonstrate the feasibility of high-precision operations, they do not consider RPO or in-space computation. They are also insufficiently domain-informed and not motivated by a realistic understanding of the security concerns that exist in space. In particular, they make assumptions about the values that satellites seek to keep private during conjunction analysis, including position, velocity, and radius values as defined by Alfano's method [2]. However, not all inputs need to be kept private in conjunction analysis. For example, the position can be known by spacecraft operators or easily determined with high precision from ground stations [44]. This causes unnecessary private evaluation, increasing overhead in resource-constrained systems where computation should be optimized. In contrast, covariance matrices and sensor-specific characteristics can reveal intrinsic properties and thus necessitate privacy preservation, as these values cannot be measured with sufficient accuracy by observation alone.

Our research differs from existing work as we present a case for private collision avoidance when satellites approach each other at close distances, where ground station control cannot be relied upon, and the operations must occur directly on the satellite. The work done to motivate the use of MPC in RPO applications is limited. Fedele et al. [29] address the need for security in RPO and conducted empirical tests, but their tests do not consider radiation-tolerant components, and they evaluated only a single MPC algorithm for the RPO problem of attitude determination. Little research has been done on the security of in-space servicing beyond discussing the need for hardware security and authorization [17]. Furthermore, MPC has advanced greatly in recent decades, making it a feasible tool for private computation in resource-constrained systems and other new environments [15, 32].

4 System and Threat Model

In space, established cryptographic practices are used to protect data 'in transit' or 'at rest', but there is currently no accepted standard for protecting data 'in use' [8]. MPC is a well-established cryptographic solution to this problem and offers distinct guarantees of privacy and correctness during computation [51]. The system we model in this paper is that of a coalition of satellites, considering cases of two or three spacecraft involved in joint computation. MPC eliminates the need for a trusted third party and computation is performed jointly, without disclosing private data. We study two fundamental space computations: an artificial potential function (APF) and a quadratic program (QP). These algorithms are general approaches that are used in spacecraft controls and can be tailored to solve a variety of space missions [60, 70, 80, 90]. In this paper, we give two examples of computations where MPC is used to secure RPO: APF to demonstrate collision avoidance and QP for multi-point inspection. We assume that participating satellites are within RPO distances, less than 500 kilometers from each other, and therefore all communication and joint computing occur directly over satellite crosslinks.

Each satellite's security model is highly mission-dependent and requires designers and operators to determine the security expectations of their system. Two types of adversarial models are relevant to RPO applications: *semi-honest* and *malicious*. The semi-honest (passive) model assumes that the computing parties follow the prescribed computation but can try to learn unauthorized information about the private data they handle during computation. The malicious (active) model allows for corrupt participants that can arbitrarily deviate from the prescribed computation and attempt to compromise correctness or privacy. There is a performance cost to ensuring greater security, as protocols in the malicious model incur heavier costs compared to protocols with semi-honest participants. We also consider two different scenarios regarding corruption, the number of parties that can be compromised during a computation. For some number of computing parties, n, a threshold, t-out-of-n, can be corrupted without information about the secret input being leaked. The *honesty majority* guarantees security for $t < n/2$ and *the dishonest majority* for $t < n$. Honest majority is generally more efficient while dishonest majority offers stronger security notions as it supports greater tolerance to corruption. The constraints on the system include bandwidth and latency due to satellite RF bands and distance from each other, material and hardware limitations, such as radiation tolerance and memory capacity, and constraints on power usage and fuel. In these applications, satellites must share

vehicle dynamics values to prevent collisions. As described in Section 3, while a satellite's location is observable, intrinsic properties of its sensors that impact navigation must be protected. Covariance matrices, in particular, are a crucial component of safe collision avoidance [79]. Furthermore, they can be used to infer satellite's capabilities, and must be protected as detailed in Section 4.2.

4.1 Security Model

Trust is a crucial part of space operations, particularly between disparate stakeholders, but even a high level of trust does not negate the need for strong data protection. For example, two nations may collaborate on missions that benefit both of them, such as with the ISS, but seek to avoid divulging certain satellite design specifications. Within the same country, multiple agencies, such as scientific agencies and military arms (e.g. NASA, U.S. Air Force), often collaborate on missions but, having different agency-wide security regulations, may require that details of their satellite's data processing or telemetry remain private [89]. In addition, private companies (e.g. SpaceX, Intelsat) are now at the forefront of space innovation and have the goals of protecting intellectual property and proprietary designs while working closely in areas of development.

Based on the security guarantees of different MPC protocols, we can draw conclusions about three cases where different MPC security models would have appropriate usage in space.

(1) **Honest majority, semi-honest model.** A scenario for this model is MPC operation within an organization, where most or all computing parties are trustworthy, but accidental leakage must be prevented.
(2) **Dishonest majority, semi-honest model.** Here is a comprised coalition of multiple stakeholders with a common mission goal. For example, organizations that are cooperative but must prevent passive adversaries from learning proprietary satellite design or operating information.
(3) **Dishonest majority, malicious model.** This would be necessary when joint computation must be done between untrustworthy or uncooperative organizations, with expectations of an attack on the protocol functionality itself.

4.2 Scenarios and Adversarial Capabilities

In general, satellite operators want to share as little information with each other as possible [48]. However, to maintain physical safety during RPO, certain information must be shared, namely position, velocity, torque, or state-of-health telemetry, such as heat or power usage, attitude, and fuel levels. The covariance matrices associated with such position, velocity, and other dynamics are typically shared in order to assess risks and maintain high precision in maneuvering.

To better explain this, we provide two common examples of RPO explored in this study: collision avoidance and multi-point inspection.

4.2.1 Collision Avoidance. A state-of-the-art method for autonomously adjusting trajectories when satellites are within 500 kilometers of each other and therefore require RPO capabilities is the artificial potential function (APF) [77], described in detail in Section 5.2.1. In contrast to relative position (determining distance between two satellites), velocity, and torque, which can be directly measured at close distances, the covariance metrics, also called *noise covariance*, are determined by factoring in noise that is inherent to the satellite's sensors and cannot be inferred without a satellite sharing this information. Additionally, stakeholders may need to keep future satellite location information unknown. Private companies have a vested interest in securing their on-orbit assets to maintain a competitive advantage, and governments see these data as a national security concern [34]. The covariance of the ephemerides can allow adversaries to predict the satellite's trajectory with high accuracy. Both of these vulnerabilities are addressed by sharing covariance matrices in a privacy-preserving manner.

4.2.2 Multi-Point Inspection. There is great motivation to advance the resilience, functionality, and sustainability of space technology through in-space service, assembly, and manufacturing (ISAM) [64]. ISAM enables mission-extension processes for OSS, refueling, and assembly through advanced robotics and integrative management technologies [98]. Orbital manufacturing facilities, or factories-in-space (FiS), would enable reduced launch and servicing costs as well as advanced development that *in situ* space conditions (particularly microgravity) offer to materials manufacturing and bioengineering [45, 57]. Privacy is a known issue for manufacturing, where the exchange of production data may be necessary for rapid feedback, but can expose intellectual property (IP) to collaborating companies [73]. Privacy is needed in ISAM, where multiple satellites owned by different organizations or countries are operating as independent critical nodes in an FiS supply chain, especially for the management of autonomous processes and feedback cycles [57]. One of such processes is multi-point inspection, a preliminary step in servicing, docking, or debris removal. The details of this computation are explained further in Section 5.2.2. Covariance matrices are also communicated in this case and should be protected for the same characterization issues as discussed in Section 4.2.1.

5 Methodology

Given our understanding of space computation and the limitations of the space environment as defined in Section 4, we present our approach to determine the feasibility of MPC in RPO settings. We first explain the environment in which the computation is executed, and then detail the computation used in our target applications.

5.1 Execution Environment

The benchmarking approach taken in this study is based on the main resource constraints that any algorithm expected to run in space will face. The primary constraint is execution time, but communication volume and number of communication rounds are also significant factors [40]. In RPO, time requirements are contingent upon algorithm refresh rates, which are mission-dependent, but we can set an upper bound to what reasonable requirements would be. We determined time thresholds using a challenge problem that uses safe reinforcement learning to dictate the autonomy constraints necessary for RPO maneuvers [75]. Conceptually, we use a "passively safe" model, which optimizes RPO calculations for both time and fuel levels. If an algorithm is called too frequently, fuel is wasted to initiate thrusters to move and combat noise. If called too infrequently, the satellites involved are put at risk. This informs several

On Board		Network	
Execution time	30 sec – 5 min	Frequency Band	S, X, Ku, Ka, optical
Memory	10 MB	Rate	10 Mbps – 1 Gbps
Examples	Dove constellation, OneWeb and SpaceX's Ku-Ka-band satellites, Kepler system, Spire's small satellites		

Table 1. Sampling of values for benchmarking in-space RPO computation and examples of small satellite missions under these constraints [56].

safety constraints imposed on a satellite from both a computational and a physical location perspective.

Although it may be possible to quickly actuate or let vehicles drift, an execution time ranging from about 30 seconds to 5 minutes tends to balance fuel efficiency and desired mission performance [75]. Therefore, our most constrained upper time bound for the execution of either application is 30 seconds, as this is the limit for the frequency with which thrusters can actuate autonomous updates [61, 74]. It is important to note that in *relative orbital dynamics*, although satellites travel at high speeds respective to the earth, they move slowly relative to each other and therefore have ample time to complete calculations [81]. Satellites also operate in an extremely compartmentalized manner with resources allocated precisely for each process [92], so science and mission objectives are segregated from operational procedures, such as a collision avoidance program. However, a satellite can divert significant power to collision avoidance if necessary while attempting to reach a safe state again, though this would be a rare case. GNC algorithms, such as collision avoidance, typically run every few minutes in the background without impacting any other processes [92]. This is important to note because the additional overhead required to incorporate data protection into one of these algorithms does not change the functionality of the satellite or hinder any other processes running on board as long as the algorithm runs within the time threshold given above [92]. Typical parameters and constraints, listed in Table 1, are based on reviews of publicly available small satellite data and design specifications [24, 56, 72, 99].

Time, power, and network resources are all limited in space, and therefore, satellites require specification and optimization for on-board processes. Although there is a range of data rates that small satellites use, the majority operate in the S- or X-band radio frequency at rates of about 10 to 150 Mbps [24, 99]. More information on satellite transmission and communications can be found in Appendix A. In our analysis, we restrict bandwidth to the transmission rate of the S-band frequency, 10 Mbps. This is the most limited rate for this class of satellites, which allows us to ensure that if we meet the timing constraints on the slowest network, our solutions will also be suitable for higher-bandwidth systems. Based on this consideration, we set the network latency to reflect signal propagation between satellites. Assuming a distance of $d = 500$ km and travel speed of light $c = 3 \times 10^8$ m/s for RF signals, we calculate a one-way latency of $t = \frac{d}{c} = 1.2$ milliseconds. In this setup, the total execution time accounts for local execution time on board, transmission time for each communication round, and signal propagation time.

5.2 RPO Computation

Much of RPO mathematics and theory define operations for two vehicles, such as docking and near-field collision avoidance, but many operations are also designed for safe collaboration between three or more vehicles. One specific operation is inspection, which cannot be effectively performed with fewer than three satellites. Relative mechanics in three dimensions requires at least three points of reference to accurately reflect given measurements. Other examples with three or more satellites include coordinated OOS, manufacturing, and formation flying [22]. We tested two- and three-party RPO scenarios to characterize how MPC serves as a privacy solution for these different operations. The first scenario is collision avoidance, which is a motivating case for two-party MPC. The second scenario is multi-point inspection of a satellite or object, which uses three or more computing parties in MPC. We carefully examine all values entered into joint computation with respect to their sensitivity. All inputs are protected unless they are fixed or can be easily determined by other satellites (e.g., coordinates). [2]

5.2.1 Collision Avoidance. Collision avoidance has been used for decades to protect satellite interoperability but is not always an in-space calculation. Standard conjunction analysis, as performed by ground stations, uses two-dimensional encounter geometry (detailed in [3]), but is insufficient for RPO scenarios. We use a GNC algorithm, an artificial potential function (APF), to perform collision avoidance calculations, as introduced in Section 4.2.1. The APF uses potential fields, a concept in robotics controls, where an object is either "attractive" or "repulsive" to a vehicle depending on its objectives, and is state-of-the-art for obstacle avoidance and docking. For collision avoidance, one satellite has repulsive potential relative to the other, and an avoidance region is defined around the other. For docking, an attractive potential is calculated for a designated region surrounding the other satellite [100]. The APF computation involves evaluating the set of equations described in [16]. It assumes relative motion (one satellite is stationary relative to the other) between two spacecraft, which are referred to as the object and the chaser [100]. The chaser is trying to reach a target position in space while avoiding the object. The variables $\mathbf{c_{pos}}$, $\mathbf{o_{pos}}$, and $\mathbf{t_{pos}}$ denote the chaser, object, and target locations, respectively, each represented as three-dimensional coordinates $(x, y, \theta) \in \mathbb{R}^3$ (values in the third dimension are expressed in radians). The APF computes the control forces necessary to reach the desired position using the gradient of a potential field, which is made up of attractive and repulsive potentials [100]. It is important to note that to travel along these gradients the direction is negative by definition as the algorithm optimizes using negative feedback, which is standard in controls literature. The attractive potential, ϕ_a, establishes a global minimum at the desired target position. It is defined as:

$$\phi_a = \frac{k_a}{2} \mathbf{r_{ct}}^T \mathbf{Q}_a \mathbf{r_{ct}}$$

where $\mathbf{r_{ct}} = \mathbf{t_{pos}} - \mathbf{c_{pos}}$ is the relative difference in the current target and chaser positions, $\mathbf{Q_a} \in \mathbb{R}^{3\times3}$ is a diagonal shaping (covariance)

[2] Bold lowercase letters denote vectors and the bold uppercase letters denote matrices.

matrix that defines an ellipsoid of the target location in space, and $k_a \in \mathbb{R}_+$ is a gain value that defines how much influence each control input has on the state variables of the satellite.

The repulsive potential, ϕ_r, defines an exclusion zone, a region in space of higher potential that the chaser should avoid, which in this case is the ellipsoid surrounding the object satellite. The relative position of the chaser to the object $\mathbf{r_{co}} = \mathbf{o_{pos}} - \mathbf{c_{pos}}$ is used to obtain the repulsive potential:

$$\phi_r = \psi e^{-\frac{\mathbf{r_{co}}^T \mathbf{N} \mathbf{r_{co}}}{\sigma}}$$

where $\mathbf{N} \in \mathbb{R}^{3\times3}$ is another diagonal shaping (covariance) matrix that represents the object satellite and contains confidence weightings to understand its various directions. ψ and σ are the height and width of the chaser in meters. The resulting total potential is:

$$\phi_{\text{tot}} = \phi_a + \phi_r$$

Using the formulas above, the gradient potential is given as:

$$\nabla\phi_{\text{tot}} = \nabla\phi_a + \nabla\phi_r = k_a \mathbf{Q_a} \mathbf{r_{ct}} - \frac{2\psi}{\sigma} e^{-\frac{\mathbf{r_{co}}^T \mathbf{N} \mathbf{r_{co}}}{\sigma}} \mathbf{N} \mathbf{r_{co}} \tag{1}$$

The chaser uses a continuous feedback control law for precise control in close proximity to another satellite, represented as the vector $\mathbf{u} \in \mathbb{R}^3$, which represents the set of control inputs applied to the satellite's propulsion system, defined as:

$$\mathbf{u} = -\mathbf{K_{acc}} \mathbf{B}^{-1} (\mathbf{c_{vel}} + \nabla\phi_{\text{tot}}) \tag{2}$$

where $\mathbf{B} \in \mathbb{R}^{3\times3}$ is a diagonal matrix representing guidance parameters that use the mass and moment of inertia of the chaser satellite, $\mathbf{K_{acc}} \in \mathbb{R}^{3\times3}$ is a positive gain matrix and $\mathbf{c_{vel}} \in \mathbb{R}^3$ is the chaser's velocity.

When running the APF using MPC, the first step is to identify which values are to be protected and which source contributes specific private values. In the context of the APF, the positions of the chaser, object, and target (i.e., $\mathbf{c_{pos}}$, $\mathbf{o_{pos}}$, $\mathbf{t_{pos}}$) are considered public because they are observable or can be approximated with reasonable accuracy. This means that their relative distances $\mathbf{r_{ct}}$ and $\mathbf{r_{co}}$ are also public. We also treat gain values k_a and $\mathbf{K_{acc}}$ as public information. The remaining values are private. In particular, they consist of proprietary information about the chaser satellite or the way it is currently moving ($\mathbf{B}$, $\mathbf{Q_a}$, ψ, σ, $\mathbf{c_{vel}}$) and proprietary information about the object ($\mathbf{N}$). The chaser could choose to keep ψ, σ, and $\mathbf{c_{vel}}$ private to make it harder for the other satellite to obtain these values, but ultimately these could be learned by repeated measurement by the object. The output $\mathbf{u}$, which helps the chaser navigate, is delivered only to the chaser, the primary computing party in this setup. We summarize this information in Table 2.

An important consideration is the privacy guarantees of this approach, specifically the amount of information that can be disclosed about sensitive inputs from the computation output. The object cannot learn any information as it does not obtain any output. We only need to evaluate the contribution of the object's private matrix $\mathbf{N}$ to the output $\mathbf{u}$. $\mathbf{N}$ is used in equation 1 during the computation of $\nabla\phi_r$, the result being used to calculate equation 2. Note that the chaser will be able to determine $\nabla\phi_r$ from the output $\mathbf{u}$ since all the components in equations 1 and 2, except $\mathbf{N}$, are contributed by the chaser and all but $\nabla\phi_r$ are linear. Furthermore, since $\mathbf{N}$ is a diagonal matrix, it is possible to remove $\mathbf{r_{co}}$ from $\mathbf{N}\mathbf{r_{co}}$ and remove $\frac{2\psi}{\sigma}$ from the gradient. The chaser can then recover the component $e^{f(\mathbf{r_{co}}, \mathbf{N}, \sigma)}\mathbf{N}$ and brute-force the elements of $\mathbf{N}$. One privacy problem is that the ratio of the elements of $\mathbf{N}$ is preserved in the output, and it is possible to search for these elements. We assume an adversary capable of characterizing the sensors on-board the satellite from the provided covariance matrix $\mathbf{N}$.

Variable	Meaning	Type
$\mathbf{c_{pos}}$, $\mathbf{o_{pos}}$, $\mathbf{t_{pos}}$	Chaser, object, and target positions, respectively	public
k_a, $\mathbf{K_{acc}}$	Gain values	public
$\mathbf{B}$	Input guidance parameters	private to chaser
$\mathbf{Q_a}$	Target covariance matrix	private to chaser
$\mathbf{c_{vel}}$	Chaser's velocity	private to chaser
ψ, σ	Chaser's height and width	private to chaser
$\mathbf{N}$	Object's covariance matrix	private to object
$\mathbf{u}$	Control force	private to chaser

Table 2. Summary of APF input values, what they designate, and whether they are public or private to one of the satellites.

Standard solutions such as differential privacy (designed to protect a record in a dataset) are not applicable to this single-input scenario [26]. Instead, we choose to modify the output, $\mathbf{u}$, to prevent recovery attacks on $\mathbf{N}$ while preserving the most significant bits of the output's precision. We add noise to $\mathbf{u}$ that will break the exact relationship between the elements of $\mathbf{N}$, but is within the error bounds of the computation. The acceptability of modifying the result is mission dependent, and factors in the safety concerns of collision avoidance as well as constraints on fuel consumption. Nevertheless, deviations in the range of 1–5% are deemed acceptable in orbital dynamics [36], as control actuation for certain thrusters can only be accomplished with this same relative accuracy [31, 97]. We determined the most significant non-zero bit of each element of the output vector and added randomly generated noise starting from a certain offset (e.g. 6 bits) from the most significant non-zero bit. By adding random perturbations to the least significant bits of the output $\mathbf{u}$, these bits become unrecoverable. This prevents the recovery of the exact value of $\nabla\phi_r$, which is necessary to calculate the exact elements of $\mathbf{N}$. That is, one can no longer carry out the exact brute force and factoring attack described above. By performing this perturbation, we induce an additive error in the recovered elements of $\mathbf{N}$ proportional to the amount of random error imposed on the elements of $\mathbf{u}$. Since $\mathbf{N}$ represents a covariance matrix characterizing the satellite's understanding of its position in space, built of input from multiple sensors [39], the space of measurable $\mathbf{N}$ matrices is effectively continuous. As such, we can minimize the threat of characterization through this perturbation without putting the physical safety of the satellite at risk. The pseudocode in Section 6.1 demonstrates this implementation.

5.2.2 Multi-Point Inspection. The second algorithm we test is a sensor fusion algorithm, which we implement for a satellite multi-point inspection scenario, as explained in section 4.2. For this work, we cast the problem as a QP, which is a class of optimizations that can be solved rapidly and reliably. QPs are highly efficient and effective in generating optimal guidance and navigation commands.

For example, this type of program was used to perform entry guidance for the Mars *Perseverance* rover in 2021 [54] and is used for controlled orbit around another object for purposes such as flyover imaging [12]. This algorithm determines a statistical model of position uncertainty to improve the robustness of proximity operations, specifically developed for spacecraft RPO in the presence of nearby obstacles or vehicles.

The optimization in this scenario minimizes or maximizes a quadratic function subject to linear constraints that reflect satellite inspection [66]. It takes multiple probabilistic inputs and produces an optimized output value. For this work, we assume that three satellites are involved in joint computation. The inputs are each satellite's measurement of the position coordinates, **pos**, of another satellite or object being inspected. Each participating satellite also enters the covariance matrix, **P**, associated with its sensors' measurement of the inspected object, into the computation. The inspection vehicle performing the computation wants to know what each of the other satellites is measuring. They also share the result with each other for the sake of physical safety. The goal of the QP is to determine the minimizing vector, **e**, which can be plugged into the quadratic function to produce the global minimum of the function. This is found by taking the QP gradient and setting it to zero. The linear quadratic term in the QP is calculated as:

$$\begin{aligned} g &= \min[(\mathbf{pos_1}-\mathbf{e})^T\mathbf{P_1^{-1}}(\mathbf{pos_1}-\mathbf{e}) + (\mathbf{pos_2}-\mathbf{e})^T\mathbf{P_2^{-1}}(\mathbf{pos_2}-\mathbf{e}) \\ &\quad + \cdots + (\mathbf{pos_n}-\mathbf{e})^T\mathbf{P_n^{-1}}(\mathbf{pos_n}-\mathbf{e})] \end{aligned} \tag{3}$$

where $\mathbf{P_1}, \ldots, \mathbf{P_n} \in \mathbb{R}^{3\times3}$ are 3×3 matrices, $\mathbf{pos_1}, \ldots, \mathbf{pos_n} \in \mathbb{R}^3$ are position vectors for three spatial coordinates, and $\mathbf{e} \in \mathbb{R}^3$ is the minimizing vector. Next, the gradient of $g(\mathbf{pos})$ is calculated and set to zero. With $n = 3$ for the three-satellite setup, we calculate:

$$\begin{aligned} \nabla g &= 2(\mathbf{P_1}^{-1} + \mathbf{P_2}^{-1} + \mathbf{P_3}^{-1})\mathbf{e} - \\ &\quad 2(\mathbf{P_1}^{-1}\mathbf{pos_1} + \mathbf{P_2}^{-1}\mathbf{pos_2} + \mathbf{P_3}^{-1}\mathbf{pos_3}) \\ &= \mathbf{M}\cdot\mathbf{e} + \mathbf{V} = 0 \end{aligned} \tag{4}$$

and solve for the minimizer by calculating:

$$\mathbf{e} = \mathbf{M^{-1}V} \tag{5}$$

Optimization produces an estimate of the true position, $\mathbf{pos}_{opt}, \mathbf{P}_{opt}$, of the inspected satellite. When applied to the MPC context, we treat the entered coordinates as public values. They would not benefit from being privatized when the satellites are close enough to obtain accurate position measurements. Covariance matrices, on the other hand, are treated as private data. The result **e** is delivered to the computing satellites. Similarly to the APF computation, we assess the protection of each party's covariance $\mathbf{P_i}$ afforded by using secure computation. From equations 4 and 5 we see that the computation of **e** involves linear operations. Recent work rigorously studied information disclosure from the output of the summation function for n inputs [6, 7]. Although our computation is more complex, we expect the high-level trends to hold because of its linearity. Suppose that the results are delivered to satellite one, which is in possession of $\mathbf{P_1}$. If it tries to target the private input of another satellite, such as $\mathbf{P_2}$, then the presence of another input, $\mathbf{P_3}$ in this case, protects the input and prevents its recovery. Other work in MPC has shown that even with a single third-party input (not belonging to the adversary or the target), information disclosure about the target is limited to a fraction of a bit and quickly decreases when more inputs are used [6, 7]. Three-satellite inspection is a typical scenario and more satellites could be used for improved precision.

6 Implementation

To evaluate the performance of the APF and QP applications, we use an MPC compiler that translates a program written in a conventional programming language extension (Python) to a secure MPC protocol. Our threat model demands the ability to execute the computation under a variety of adversarial settings, ranging from those tolerating a semi-honest adversary controlling a minority of the participants to those resilient to a malicious adversary that controls a majority of the participants. MP-SPDZ [40] is currently one of the most comprehensive MPC compilers, which we choose to evaluate our RPO computations. The MP-SPDZ suite provides a rich and varied set of protocols that have different properties and settings, so a characterization of these well-constructed protocols is an important step forward before making (potentially incorrect) design decisions related to developing new cryptographic protocols from scratch.

Almost all of the implemented protocols in MP-SPDZ are based on secret sharing, operating over a finite ring or a finite fields (which is applicable to both honest majority and dishonest majority settings). The number of computation rounds varies depending on the computation being performed, and lowering the round complexity is of central importance to performance. It is worth noting that with MP-SPDZ, computation and communication are divided into offline and online work, with the former being input-independent generation of random bits, multiplication triples, etc. With the APF's collision avoidance computation, preplanned computation is not possible, as collisions are not planned for, but the QP computation can be prepared in advance. This would be the case for scheduled inspection or repairs, where participants can execute the offline phase ahead of time when they are at a communicable distance, reducing execution time for the online phase.

6.1 Secure Computation Algorithms

We translate the APF and QP computations from Sections 5.2.1 and 5.2.2 respectively into secure computations optimized to reduce cost. We operate directly on vectors and matrices to maximize parallelism. Both programs operate over real numbers, and the corresponding computation would be implemented using floating-point arithmetic in a conventional program. However, MP-SPDZ supports only fixed-point arithmetic. We therefore implemented the programs in an extension of Python and evaluated the MPC-based implementation result for correctness on a typical range of inputs for our programs of interest. The fixed-point precision in our MP-SPDZ programs was set to achieve accuracy comparable to that of floating-point computation in conventional programs. For APF, bit length was set to 33 bits for the integer and 30 bits for the fractional parts, and for QP, to 35 integer bits and 36 fractional bits. In Algorithms 1 and 2, the vectors are in bold and are of size 3. All matrices were diagonal and are represented as vectors of their diagonal elements. For vectors containing coordinates (e.g. $\mathbf{c_{pos}}$, $\mathbf{o_{pos}}$, $\mathbf{t_{pos}}$ in the APF), the values at positions 0 and 1 refer to the coordinates x and y, respectively, and the value at position 2 refers to the angle. In Algorithm 1, calculation of the attractive

Algorithm 1 Artificial Potential Function

Public parameters: $k_a \in \mathbb{R}, \mathbf{K_{acc}} \in \mathbb{R}^3$ (gain and shaping factors) All matrices ($\mathbf{K_{acc}}, \mathbf{Q_a}, \mathbf{N}, \mathbf{B_{inv}}$) are assumed to be diagonal and replaced with vectors of their diagonal elements

1: **procedure** APF($\mathbf{c_{pos}}, \mathbf{o_{pos}}, \mathbf{t_{pos}} \in \mathbb{R}^3, [\![\mathbf{c_{vel}}]\!], [\![\mathbf{Q_a}]\!], [\![\mathbf{N}]\!], [\![\mathbf{B_{inv}}]\!] \in \mathbb{R}^3, [\![\sigma]\!], [\![\psi]\!] \in \mathbb{R}$) ▷ coordinates $\mathbf{c_{pos}}$, $\mathbf{o_{pos}}$, and $\mathbf{t_{pos}}$ are public; private $\mathbf{N}$ is contributed by the object; private $\mathbf{c_{vel}}$, $\mathbf{Q_a}$, $\mathbf{B_{inv}}$, σ, and ψ are contributed by the chaser
2: $\mathbf{rCt} \leftarrow \mathbf{t_{pos}} - \mathbf{c_{pos}}$ ▷ chaser-target distance
3: **if** $(\mathbf{rCt}[2] > \pi)$ **then** ▷ wrap angle between $-\pi$ and π
4: $\mathbf{rCt}[2] \leftarrow \mathbf{rCt}[2] - 2\pi$
5: **else if** $(\mathbf{rCt}[2] < -\pi)$ **then**
6: $\mathbf{rCt}[2] \leftarrow \mathbf{rCt}[2] + 2\pi$
7: **end if**
8: $\mathbf{rCo} \leftarrow \mathbf{o_{pos}} - \mathbf{c_{pos}}$ ▷ chaser-object distance
9: **if** $(\mathbf{rCo}[2] > \pi)$ **then** ▷ wrap angle between $-\pi$ and π
10: $\mathbf{rCo}[2] \leftarrow \mathbf{rCo}[2] - 2\pi$
11: **else if** $(\mathbf{rCo}[2] < -\pi)$ **then**
12: $\mathbf{rCo}[2] \leftarrow \mathbf{rCo}[2] + 2\pi$
13: **end if**
14: $[\![\nabla\phi_A]\!] \leftarrow (k_a \cdot \mathbf{rCt}) \cdot [\![\mathbf{Q_a}]\!]$ ▷ attractive potential gradient
15: $[\![x]\!] \leftarrow (\mathbf{rCo} \cdot \mathbf{rCo}) \odot [\![\mathbf{N}]\!]$
16: $[\![\text{constVio}]\!] = 0$
17: **if** $([\![x]\!] < 1)$ **then**
18: $[\![\text{constVio}]\!] = 1$
19: **end if**
20: $[\![x]\!] \leftarrow -[\![x]\!]/[\![\sigma]\!]$
21: $[\![\nabla\phi_R]\!] \leftarrow 2[\![\psi]\!]/[\![\sigma]\!] \cdot e^{[\![x]\!]} \cdot (\mathbf{rCo} \cdot [\![\mathbf{N}]\!])$ ▷ repulsive potential gradient
22: $[\![\mathbf{u}]\!] \leftarrow (-\mathbf{K_{acc}}) \cdot [\![\mathbf{B_{inv}}]\!] \cdot ([\![\mathbf{c_{vel}}]\!] + [\![\nabla\phi_A]\!] + [\![\nabla\phi_R]\!])$ ▷ compute control force
23: **for** $i \in \{0, 1, 2\}$ **do**
24: $[\![\mathbf{m}[i]]\!] \leftarrow \text{MSNZB}([\![\mathbf{u}[i]]\!])$
25: $[\![\mathbf{u}[i]]\!] \leftarrow [\![\mathbf{u}[i]]\!] + (\text{Rand}() \gg (\text{bitlength} - [\![\mathbf{m}[i]]\!] + \text{offset}))$
26: **end for**
27: **return** $[\![\mathbf{u}]\!], [\![\text{constVio}]\!]$
28: **end procedure**

Algorithm 2 Quadratic Program

1: **procedure** QP($\mathbf{pos_1}, \mathbf{pos_2}, \mathbf{pos_3} \in \mathbb{R}^3, [\![\mathbf{P_1}^{-1}]\!], [\![\mathbf{P_2}^{-1}]\!], [\![\mathbf{P_3}^{-1}]\!] \in \mathbb{R}^{3\times3}$) ▷ measured coordinates $\mathbf{pos_1}, \mathbf{pos_2}, \mathbf{pos_3}$ are public; private inverse covariance matrix $[\![\mathbf{P}_i^{-1}]\!]$ is contributed by participant i
2: $[\![\mathbf{M}]\!] \leftarrow 2([\![\mathbf{P_1}^{-1}]\!] + [\![\mathbf{P_2}^{-1}]\!] + [\![\mathbf{P_3}^{-1}]\!])$
3: $[\![\mathbf{v}]\!] \leftarrow 2([\![\mathbf{P_1}^{-1}]\!]\mathbf{pos_1} + [\![\mathbf{P_2}^{-1}]\!]\mathbf{pos_2} + [\![\mathbf{P_3}^{-1}]\!]\mathbf{pos_3})$
4: $[\![\mathbf{e}]\!] \leftarrow [\![\mathbf{M}]\!]^{-1}[\![\mathbf{v}]\!]$
5: **return** $[\![\mathbf{e}]\!]$ ▷ position uncertainty
6: **end procedure**

potential gradient $\nabla\phi_A$ is performed on line 14, while the repulsive potential gradient $\nabla\phi_R$ is calculated on lines 15–21. Note that the computation on lines 16–19 checks whether the values are within the expected bounds and the result is reported as part of the output. The total gradient and corresponding control force are determined on line 22, with the latter reported to the chaser as the output. With MPC, the most expensive operation is the exponentiation on line 21. Lines 23–26 correspond to the addition of noise to the output vector $\mathbf{u}$, as explained in Section 5.2.1. Here, MSNZB corresponds to the computation of the position of the most significant non-zero bit of its argument, with the least significant bit being at position 0. We implement MSNZB functionality in MP-SPDZ by performing bit decomposition of the input, followed by prefix OR and a sum. The next line generates a random element by calling to function Rand. The random element is shifted to the right by a private number of bits to leave the offset most significant non-zero bits of $\mathbf{u}[i]$ unchanged while obfuscating other bits.

Algorithm 2 provides the details of the QP calculation. As before, the positions of the satellites are treated as public information, while their proprietary details, represented as covariance matrices, are private inputs. The parties invert their covariance matrices prior to entering them into the computation to reduce the overhead of secure computation. The computation consists of several addition and multiplication operations, with the matrix inversion on line 4 being the most costly operation that impacts the overall runtime. Although MP-SPDZ provides a numerical approximation algorithm to calculate the inverse of a matrix, it was unstable, even while using relatively small inputs of the order of 10^1. Therefore, we implement a 3×3 matrix inversion algorithm from the user program by dividing the adjoint matrix with the determinant.

7 Performance Evaluation

7.1 Experimental Setup

To assess the practicality of modeling our algorithms in space, we examined processors that have been tested for the harsh space environment. Ideally, hardware should be commercial-off-the-shelf (COTS) to reduce financial costs for stakeholders, but all electronic parts that run in space must tolerate a high radiation environment [88]. Based on radiation testing, several types of NVIDIA embedded boards tolerate the space environment very well [55, 85, 86]. Specifically, NVIDIA Jetson Nano embedded boards are predicted to survive a minimum of 1.5 years in LEO with some aluminum shielding [86]. Therefore, our algorithms are tested on three Jetson Nanos to effectively emulate two- and three-party MPC. These boards feature four ARM Cortex-A57 CPUs with a 1.47 GHz clock speed and a 1 Gbps Ethernet port. These devices are connected on a LAN with an induced one-way latency of 1.2 ms and a bandwidth of 10 Mbps to reflect the network conditions in space discussed in Section 5.1. This models inter-satellite links at short RPO distances with an average two-way latency of 3.757 ms between each device and a bandwidth of 9.57 Mbps. Each experiment was executed 10 times, and the average runtime is reported. Of the 10 executions, there was virtually no variance in any of the benchmarks so we did not increase this number or elaborate on this further.

7.2 MPC Configurations

In this section, we provide an overview of the techniques implemented in MP-SPDZ, their corresponding security models, and properties relevant to the scope of this work. We primarily focus on secret sharing-based techniques, as explained in the following.

Honest Majority. This setting requires that fewer than half of the participants be corrupt, the collusion threshold being $t < n/2$. These protocols offer substantially faster performance compared to dishonest majority techniques, but come at the expense of a weaker threat model. There are two types of commonly used secret sharing techniques in this model: Shamir secret sharing [84] and replicated secret sharing [52]. The former requires that each participant maintains a single share and is easy to setup with any number of participants. The advantage of replicated secret sharing is that it can be instantiated over a ring, specifically ring $\mathbb{Z}_{2^k}$ for some k, which allows the use of native CPU instructions for working with shares and leads to a significant speedup for local computation. Shamir secret sharing, however, can only be instantiated over a field, meaning that it has to use slower local operations. The disadvantage of replicated secret sharing is that each party maintains and computes with multiple shares, the number of which grows exponentially with the number of computing parties n. For that reason, MP-SPDZ implements replicated secret sharing protocols only for a special case of $n = 3$ (with $t = 1$).

MP-SPDZ offers three options for the honest majority setting: Shamir, replicated ring, and replicated field. Although the replicated field appears to combine the disadvantages of both Shamir and replicated secret sharing and would not be competitive, we find in some experiments (Section 7.3) that the replicated field was the best performing option. To understand this, we ran micro-benchmarks to examine the elementary protocols beyond the secret sharing itself. The answer lies in more efficient protocols. Recall that our computation makes substantial use of fixed-point arithmetic. One of the basic operations, fixed-point multiplication, invokes integer multiplication followed by truncation. As only one special case of replicated secret sharing is implemented, it permits the use of very efficient truncation that avoids random bit generation (used with general-purpose truncation and Shamir secret sharing) and results in efficient performance.[3] In addition, there are more protocol options for computation over a field, which rely on the properties of fields not available for rings, and leads to improved performance for field-based protocols. This discussion is in the context of semi-honest security. MP-SPDZ also offers three options to maintain security in the presence of a malicious adversary:

- Transformation of Lindell and Nof's replicated secret sharing [52], referred to as *mal* in protocol specification;
- The same as [52] except multiplications are executed optimistically and checked later, referred to as *ps*, post-sacrifice;
- Transformation based on work of [14] and [1], with SPDZ-like MACs, referred to as *sy*.

Transformations execute the computation twice in parallel: one execution is performed on the original inputs and another is done on the same inputs multiplied by a large random value r, unknown to the parties. The two branches are checked for consistency before any results can be disclosed.

Additional notes on other aspects of the MP-SPDZ implementation are included in Appendix B.

Dishonest Majority. These protocols based on secret sharing use additive secret sharing over fields or rings. They often follow the high-level structure introduced in SPDZ [23] that emphasizes a fast online phase and delegated input-independent computation, such as multiplication triples and random bit generation, to the offline phase. In this setting, additional tools beyond secret sharing are required. The implemented protocols are the following:

- *mascot*, a successor to maliciously secure SPDZ that uses OT to generate multiplication triples [41] instead of somewhat homomorphic encryption. Then *semi* is the MASCOT-based protocol in the semi-honest model with the protection against malicious adversaries removed.
- *spdz2k*, a successor to SPDZ that adopts its techniques to a ring $\mathbb{Z}_{2^k}$ setting to take advantage of fast operations [19]. *semi2k* is its semi-honest variant, which only compiled for the two-party APF program.
- *hemi* is a semi-honest variant of another SPDZ-based protocol that uses partially homomorphic encryption for precomputation [42].
- *soho* is a semi-honest SPDZ-variant that uses somewhat homomorphic encryption [42] and performs similar to hemi while improving with a higher number of parties;
- Lastly, semi-honest *temi* is an adaption of a threshold homomorphic encryption-based protocol [20] to a newer type of partially homomorphic encryption, described in [43].[4]

Parameter tuning. There are additional settings that affect performance, one of which is the use of extended doubly authenticated bits (edaBits) [27], a technique to translate between binary and arithmetic data types in MPC, for random bit generation. This typically reduces communication, but can increase the number of rounds. EdaBits were previously shown to reduce runtime only in programs with a sufficient degree of parallelism [5]. The use of edaBits did not lead to faster performance for our programs, so they were not used. We also experimented with the batch size for precomputation, looking for a balance between the number of rounds and other resource usage. Lastly, there is a trade-off between the compile time and runtime to optimize the program being compiled, and we steer it in favor of a faster runtime.

7.3 Results

Two-party APF. We present our evaluation results starting with the two-party APF computation that assumes a dishonest majority. We show the total execution time (that includes the communication time), the amount of communication data exchanged by Party 0, and the number of communication rounds reported by MP-SPDZ. As the total cost will determine the suitability of the protocols for deployment, particularly for the APF where precomputation is not feasible, we display the overall costs in Figure 1.

We see that *hemi* has the best total time of 7.38 seconds and also the lowest total communication (Figure 1). Conversely, *soho* features the lowest number of rounds, but more data is exchanged, contributing to a larger overall runtime. Among maliciously secure protocols, *spdz2k* is faster than *mascot* with a total computation time of 965 seconds and lower communication, but exceeds our

[3] All of these fixed-point protocols are based on probabilistic truncation.

[4] There are additional MP-SPDZ protocols based on binary circuits, namely BMR and Yao's garbled circuits. However, we were unable to compile our programs using those protocols and do not include them in the evaluation. Specifically, certain data types (e.g., constant integer and fixed-point variables) and operations (e.g., the dot product) our programs use are not supported by the current implementations of those protocols.

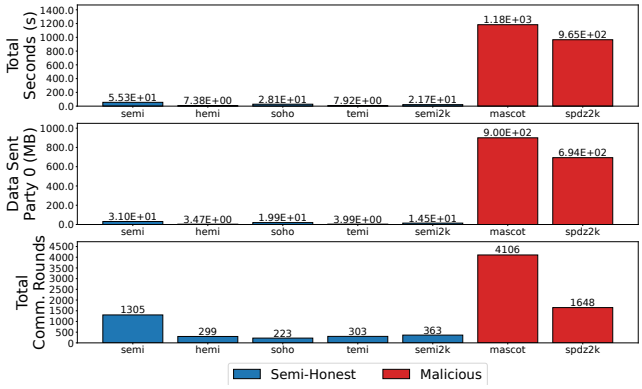


Figure 1. APF dishonest majority performance with direct communication, showing total costs.

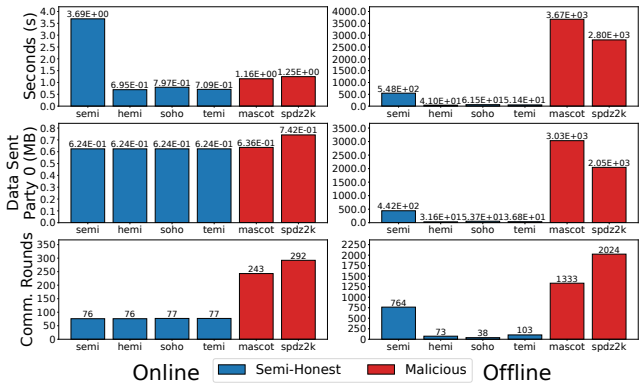


Figure 2. QP dishonest majority performance with direct communication, showing online and offline costs.

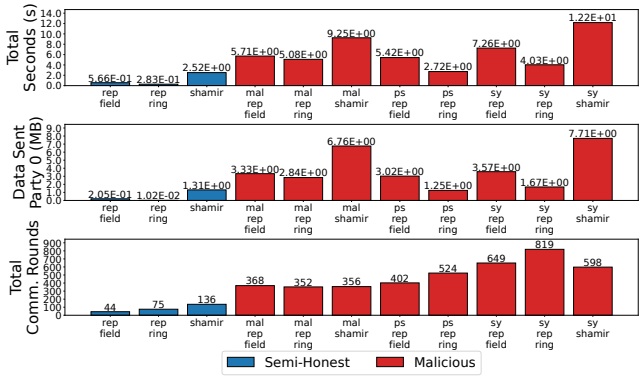


Figure 3. QP honest majority performance with direct communication, showing total costs.

desired 5-minute threshold and thus are not yet able to meet the necessary time constraints needed to guarantee RPO safety.

Three-party QP. In the three-party scenario with the QP, both dishonest and honest majority settings apply. Figure 2 shows the results for the dishonest majority setting with direct communication, showing online and offline costs. Recall that *semi2k* is not available in the three party setup. Now *hemi* shows the best performance among the semi-honest protocols, with a total time of 69.5 seconds. *semi* stands out as being significantly slower than our semi-honest protocols and is the only protocol based on oblivious transfer. This was not previously the case in the two party case of the APF and shows benefits of using homomorphic encryption techniques with three parties. Among maliciously secure protocols, the relative performance of *spdz2k* and *mascot* did not change, with spdz2k being faster. Performance in the honest majority setting is shown in Figure 3. In this case, the offline cost no longer dominates the performance so we show the combined time. The fastest protocol in the semi-honest setting is *rep-ring* with a total execution time of 0.28 sec and communication data of only 10 KB. In the malicious model, the best performing protocol is *ps-rep-ring* with execution time of 2.7 sec. Consistent with semi-honest protocols, the replicated ring variant performs the best within each group of maliciously secure protocols *mal*, *ps*, and *sy*. Performances of both semi-honest and maliciously secure protocols are within the acceptable bound for the honest majority setting.

8 Discussion

Understanding the Results. The results of our study demonstrate that MPC protocols can indeed be effectively built into satellite applications to support in-space data privacy. Given the complexity of many available MPC protocols and their variants, our results confirm that the best performing MPC protocol varies for a given satellite application and threat model. Therefore, it is important to fully understand the constraints of the environment and perform thorough evaluations under those constraints. Our evaluation shows that the primary factors in determining the best-performing MPC configuration for each in-space application are 1) the data types (e.g., fixed-point, floating-point) and types of operations and algorithms involved (e.g., arithmetic operations or complex iterative algorithms for exponentiation or matrix inversion), 2) the security requirements and expected adversarial conditions of the mission, and 3) the number of parties involved. However, the most significant factor that impacts the runtime of relatively small programs similar to those evaluated in this work is the adversarial threat model. For a given program such as the QP, switching from the honest majority setting to the dishonest majority increases the cost by orders of magnitude. Similarly, the gap between semi-honest and malicious security is not small, particularly in the dishonest majority setting. Recall that the offline phase is brief for the honest majority as it does not require the complex cryptographic mechanisms of the dishonest majority. We demonstrate that the performance of the QP in the honest majority setting, both semi-honest and malicious, is within the necessary constraints. In addition, the performance of the QP and APF programs in the semi-honest model in the dishonest majority setting also meets the requirements. This suggests that to meet the operational constraints under the stringent dishonest majority malicious model, new advances and custom protocols of improved efficiency are needed.

Limitations and Future Work. We tested both semi-honest and maliciously secure protocols, finding that only the malicious model in the dishonest majority setting exceeds operational constraints. This indicates that, given a dishonest majority, the primary

usefulness of MPC in space coalition algorithms would be in the semi-honest setting. For an honest majority, both malicious and semi-honest protocols perform within the functional requirements, allowing both of these settings to be options for consideration. Future work could include further testing and optimization for the malicious dishonest majority setting. Additionally, further understanding of information disclosure under repeated executions of an algorithm (e.g. APF or QP) would allow for stronger security assumptions or limitations to be determined and is another direction for future work.

9 Conclusion

In this paper, we address privatized space-segment satellite computation, specifically RPO, which is becoming an increasingly necessary domain for cybersecurity research. We presented secure RPO algorithms that consider space dynamics and use MPC to prevent the leakage of covariance matrices. An approach such as this is necessary to prevent satellite characterization attacks, and thus protect proprietary design and mission information. Additionally, we evaluated the disclosure of private information from each algorithm's output and provided the necessary mitigation to limit this leakage. We find that MPC is viable for assuring data privacy of in-space operations, but that the environment must be approached carefully. The best MPC technique for a given scenario in space is dependent on the problem being solved and requires an understanding of the types of operations that are performed. Our simulation results successfully identify key factors that must be considered when applying MPC to RPO based on specific mission requirements. We demonstrate that while accounting for hardware, networking, and computational costs in space, privacy-preserving collision avoidance can execute in 7.38 seconds in the semi-honest setting with a dishonest majority of participants. The quadratic program demonstrated an execution time of less than 0.3 seconds in the semi-honest setting, and less than 2.7 seconds in the malicious setting. Although dishonest majority malicious security settings would require new MPC advances in order to be realized in space, as they exceeded the 30-second lower time limit, the others, especially in honest majority settings, can be supported by off-the-shelf frameworks such as MP-SPDZ. The growing need for privacy solutions in space and the potential for new MPC technologies that can be customized for this domain demonstrate an emerging area of research rich with future opportunities.

Acknowledgments

The authors thank Marcel Keller for valuable discussions and suggestions regarding the MP-SPDZ protocols. This work was supported in part by Air Force Office of Scientific Research award FA9550-19-1-0169, US National Science Foundation grant CNS-2213057, and by a University of Florida Graduate Student Preeminence Award. Any opinions, findings, and conclusions or recommendations expressed in this publication are those of the authors and do not necessarily reflect the views of the funding sources.

References

[1] Mark Abspoel, Anders Dalskov, Daniel Escudero, and Ariel Nof. 2021. An Efficient Passive-to-Active Compiler for Honest-Majority MPC over Rings. In *ACNS*.

[2] Salvatore Alfano. 2005. A Numerical Implementation of Spherical Object Collision Probability. *The Journal of the Astronautical Sciences* (2005).

[3] Kyle Alfriend, Maruthi Akella, Joseph Frisbee, James Foster, Deokjin Lee, and Matthew Wilkins. 1999. Probability of Collision Error Analysis. *Space Debris* (1999).

[4] Space Sustainability Rating Association. 2025. *Data Sharing*. Technical Report. https://spacesustainabilityrating.org/the-rating/modules-data-sharing/.

[5] Alessandro N. Baccarini, Marina Blanton, and Chen Yuan. 2023. Multi-Party Replicated Secret Sharing over a Ring with Applications to Privacy-Preserving Machine Learning. In *PoPETs*.

[6] Alessandro N. Baccarini, Marina Blanton, and Shaofeng Zou. 2024. Understanding Information Disclosure from Secure Computation Output: A Study of Average Salary Computation. In *CODASPY*.

[7] Alessandro N. Baccarini, Marina Blanton, and Shaofeng Zou. 2025. Understanding Information Disclosure from Secure Computation Output: A Comprehensive Study of Average Salary Computation. *TOPS* (2025).

[8] Brandon Bailey, Ryan J. Speelman, Prashant A. Doshi, Nicholas C. Cohen, and Wayne A. Wheeler. 2019. Defending Satellites in the Cyber Domain. https://csps.aerospace.org/sites/default/files/2021-08/Bailey_DefendingSpacecraft_11052019.pdf. Accessed: 11-16-2025.

[9] David Barnhart, Rahul Rughani, Jeremy Allam, Brian Weeden, Frederick Slane, and Ian Christensen. 2018. Using Historical Practices to Develop Safety Standards for Cooperative On-Orbit Rendezvous and Proximity Operations. In *69th International Astronautical Congress*.

[10] Georg Baselt, James Pavur, Ivan Martinovic, Martin Strohmeier, and Vincent Lenders. 2022. Security and Privacy Issues of Satellite Communication in the Aviation Domain. https://doi.org/10.23919/CyCon55549.2022.9811060

[11] Leandra Bernstein. 2022. Inter-Satellite Links Are Making Space Networks a Reality. https://www.kratosdefense.com/constellations/articles/inter-satellite-links-are-making-space-networks-a-reality.

[12] Stefano Di Cairano, Hyeongjun Park, and Ilya Kolmanovsky. 2012. Model Predictive Control Approach for Guidance of Spacecraft Rendezvous and Proximity Maneuvering. *Int. J. Robust Nonlinear Control* (2012).

[13] Shkelzen Cakaj, Werner Keim, and Krešimir Malarić. 2007. Communications duration with low earth orbiting satellites. In *ARP*.

[14] Koji Chida, Daniel Genkin, Koki Hamada, Dai Ikarashi, Ryo Kikuchi, Yehuda Lindell, and Ariel Nof. 2018. Fast Large-Scale Honest-Majority MPC for Malicious Adversaries. In *CRYPTO*.

[15] Joseph I. Choi, Kevin R. B. Butler, and Bela Genge. 2019. Secure Multiparty Computation and Trusted Hardware: Examining Adoption Challenges and Opportunities. *Sec. and Commun. Netw.* (2019).

[16] W. H. Clohessy and R. S. Wiltshire. 1960. Terminal Guidance System for Satellite Rendezvous. *Journal of the Aerospace Sciences* (1960).

[17] CONFERS. 2024. *Best Practices, Functional Requirements, and Norms for In-space Servicing, Assembly, and Manufacturing (ISAM) Power and Data Interfaces*. Technical Report. https://cdn.ymaws.com/satelliteconfers.org/resource/resmgr/confers_publications/aiaa-proposal-form-and-confe.pdf

[18] Daniele Coppola, Harshad Sathaye, Giovanni Camurati, and Srdjan Capkun. 2025. On the Privacy of LEO Two-Way-Ranging. In *2025 Security for Space (3S) Conference*.

[19] R. Cramer, I. Damgård, D. Escudero, P. Scholl, and C. Xing. 2018. SPDZ2k: Efficient MPC mod 2k for Dishonest Majority. In *CRYPTO*.

[20] Ronald Cramer, Ivan Damgård, and Jesper Buus Nielsen. 2001. Multiparty Computation from Threshold Homomorphic Encryption. In *EUROCRYPT*.

[21] Benjamin Cyr, Yan Long, Takeshi Sugawara, and Kevin Fu. 2023. Position Paper: Space System Threat Models Must Account for Satellite Sensor Spoofing. In *SpaceSec*.

[22] Chris Daehnick, Isabelle Klinghoffer, Ben Maritz, and Bill Wiseman. 2020. *Large LEO Satellite Constellations: Will it be Different This Time*.

[23] Ivan Damgård, Marcel Keller, Enrique Larraia, Valerio Pastro, Peter Scholl, and Nigel P. Smart. 2013. Practical Covertly Secure MPC for Dishonest Majority – Or: Breaking the SPDZ Limits. In *ESORICS*.

[24] Inigo del Portillo, Bruce Cameron, and Edward Crawley. 2019. A Technical Comparison of Three Low Earth Orbit Satellite Constellation Systems to Provide Global Broadband. *Acta Astronautica* (2019), 123 – 135.

[25] Digital Frontlines. 2023. *Cybersecurity in Outer Space*. https://digitalfrontlines.io/2023/08/23/cybersecurity-in-outer-space/ Accessed: 2025-03-25.

[26] Cynthia Dwork. 2006. Differential Privacy. In *Automata, Languages and Programming*.

[27] Daniel Escudero, Satrajit Ghosh, Marcel Keller, Rahul Rachuri, and Peter Scholl. 2020. Improved Primitives for MPC over Mixed Arithmetic-Binary Circuits. In *CRYPTO*.

[28] Gregory Falco. 2018. Cybersecurity Principles for Space Systems. *Journal of Aerospace Computing, Information and Communication* (2018).

[29] Caroline Fedele, Kevin Butler, Christopher Petersen, and Tyler Lovelly. 2024. Protecting Satellite Proximity Operations via Secure Multi-Party Computation. In *AIAA SCITECH*.

[30] US Space Force. 2023. *Comprehensive Strategy for the Space Force.* Technical Report. Department of the Air Force.

[31] Jason D. Frieman, Jon Mackey, Hani Kamhawi, Peter Y. Peterson, and Richard R. Hofer. 2021. Risk-based Methodology for the Determination of Hall Thruster Performance Specifications. In *AIAA Propulsion and Energy Forum.*

[32] Idoia Gamiz, Cristina Regueiro, Oscar Lage amd Eduardo Jacob, and Jasone Astorga. 2024. Challenges and Future Research Directions in Secure Multi-Party Computation for Resource-Constrained Devices and Large-Scale Computations. *Int. J. Inf. Secur.* (2024).

[33] Felix Hanke, Johannes Unruh, Michael Karl, and Frank Köster. 2024. Secure Dataspace Approach for Interorbital Satellite Links. In *AIAA SCITECH.*

[34] Brett Hemenway, Steve Lu, Rafail Ostrovsky, and William Welser IV. 2016. High-precision Secure Computation of Satellite Collision Probabilities. In *SCN.*

[35] Caleb Henry. 2020. *Northrop Grumman's MEV-1 servicer docks with Intelsat satellite.* https://spacenews.com/northrop-grummans-mev-1-servicer-docks-with-intelsat-satellite/ Accessed: 11-13-2025.

[36] Richard S. Zappulla II. 2017. *Experimental Evaluation Methodology for Spacecraft Proximity Maneuvers in a Dynamic Environment.* Ph. D. Dissertation.

[37] Kyle W. Ingols. 2017. Design for Security: Guidelines for Efficient, Secure Small Satellite Computation. In *IMS.*

[38] Liina Kamm and Jan Willemson. 2015. Secure Floating Point Arithmetic and Private Satellite Collision Analysis. *Int. J. Inf. Secur.* (2015).

[39] Zhigui Kang, Srinivas Bettadpur, Peter Nagel, Himanshu Save, S. Poole, and Nadége Pie. 2020. GRACE-FO precise orbit determination and gravity recovery. *J Geod* (2020).

[40] Marcel Keller. 2020. MP-SPDZ: A Versatile Framework for Multi-Party Computation. In *ACM SIGSAC Conference on Computer and Communications Security.*

[41] Marcel Keller, Emmanuela Orsini, and Peter Scholl. 2016. MASCOT: Faster Malicious Arithmetic Secure Computation with Oblivious Transfer. In *CCS.*

[42] Marcel Keller, Valerio Pastro, and Dragos Rotaru. 2018. Overdrive: Making SPDZ Great Again. In *EUROCRYPT.*

[43] Marcel Keller and Ke Sun. 2022. Secure Quantized Training for Deep Learning. In *39th International Conference on Machine Learning.*

[44] John A Kennewell and Ba-Ngu Vo. 2013. An overview of space situational awareness. In *Proceedings of the 16th International Conference on Information Fusion.*

[45] Kevin Engelbert and Lynn Harper. 2023. *In Space Production Applications (InSPA): Ensuring U.S. Leadership in Advanced Materials & Manufacturing in LEO.* NASA. https://www.nasa.gov/wp-content/uploads/2022/05/in_space_production_applications_overview_-_may_2022.pdf Accessed: 11-13-2025.

[46] Syed Khandker, Krzysztof Jurczok, and Christina Pöpper. 2024. COSPAS Search and Rescue Satellite Uplink: A MAC-Based Security Enhancement. In *SpaceSec.*

[47] Mitchell Kirshner. 2023. Model-Based Systems Engineering Cybersecurity for Space Systems. *Aerospace* (2023).

[48] Clark W. Lackert. 2021. *IP in Outer Space: The Next Frontier.*

[49] Svenja Lage, Felicitas Hörmann, Felix Hanke, and Michael Karl. 2025. Collision Risk Analysis for LEO Satellites with Confidential Orbital Data.

[50] Lucas Laursen. 2023. *Satellite Signal Jamming Reaches New Lows.*

[51] Yehuda Lindell. 2020. Secure Multiparty Computation (MPC). *Commun. ACM* (2020).

[52] Yehuda Lindell and Ariel Nof. 2017. A Framework for Constructing Fast MPC over Arithmetic Circuits with Malicious Adversaries and an Honest-Majority. In *CCS.*

[53] Pierre Lionnet. 2024. *SpaceX and the Categorical Imperative to Achieve Low Launch Cost.*

[54] Xu Liu, Shuang Li, and Zhenbo Wang. 2023. Mars Entry Tracking Guidance via Quadratically Constrained Quadratic Programming and Pseudospectral Method. *AIAA Journal of Spacecraft and Rockets* (2023).

[55] Tyler M. Lovelly, Jesse K. Mee, James C. Lyke, Andrew C. Pineda, Kenneth D. Bole, and Robert D. Pugh. 2019. Evaluating Commercial Processors for Spaceflight with the Heterogeneous On-Orbit Processing Engine. In *IEEE Aerospace Conference.*

[56] Nelson Malaguti. 2024. *Handbook on Small Satellites.*

[57] Harsha Malshe, Salil Bapat, John Vickers, and Ajay P. Malshe. 2023. Factories-in-Space for Servicing, Assembly, & Manufacturing. *Manufacturing Letters* (2023).

[58] Damiano Marsili, Nicolo Boschetti, Nathaniel Gordon, Yanni Nikas, Will Leger, Malcolm Joyce, and Gregory Falco. 2023. Slipping Through Attackers' Fingers: Fast Neutron Communications for Space Cybersecurity. In *IEEE Aerospace Conference.*

[59] Jonathan McDowell. 2025. Active Satellites in Orbit.

[60] Daniel Morgan, Soon-Jo Chung, and Fred Y Hadaegh. 2014. Model predictive control of swarms of spacecraft using sequential convex programming. *Journal of Guidance, Control, and Dynamics* (2014), 1725–1740.

[61] Robert Muldrow, Channing Ludden, and Christopher Petersen. 2025. Comparison of Forced and Unforced Rendezvous, Proximity Operations, and Docking Under Model Mismatch. In *AAS SFM.*

[62] NASA. 2025. In-Space Servicing, Assembly, and Manufacturing (ISAM). https://www.nasa.gov/nexis/isam/ Accessed: 2025-04-01.

[63] NASA. 2025. *Rendezvous and Proximity Operations - Technology Development.* Technical Report. https://techport.nasa.gov/projects/125706 Accessed: 2025-04-01.

[64] National Science and Technology Council. 2022. *In-Space Servicing, Assembly, And Manufacturing National Strategy.* https://www.whitehouse.gov/wp-content/uploads/2022/04/04-2022-ISAM-National-Strategy-Final.pdf

[65] Ehsan Nekouei, Mikael Skoglund, and Karl H. Johansson. 2018. Privacy of Information Sharing Schemes in a Cloud-based Multi-sensor Estimation Problem. https://arxiv.org/abs/1802.00684

[66] Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization.*

[67] Department of the Air Force. 2024. *U.S. Space Force Commercial Space Strategy.* Technical Report.

[68] Office of U. S. Government Accountability. 2022. *Large Constellations of Satellites: Mitigating Environmental and Other Effects.* Technical Report.

[69] Carmen Pardini and Luciano Anselmo. 2021. Evaluating the Impact of Space Activities in Low Earth orbit. *Acta Astronautica* (2021).

[70] Hyeongjun Park, Stefano Di Cairano, and Ilya Kolmanovsky. 2011. Linear quadratic model predictive control approach to spacecraft rendezvous and docking. In *AAS SFM.*

[71] Patrick Lin. 2024. *Cyberattacks in Space: Growing Threats to Satellites and Space Systems.* https://phys.org/news/2024-07-cyberattacks-space.html

[72] Joseph Pelton and Scott Madry. 2020. *Handbook of Small Satellites.*

[73] Jan Pennekamp, Erik Buchholz, Yannik Lockner, Markus Dahlmanns, Tiandong Xi, Marcel Fey, Christian Brecher, Christian Hopmann, and Klaus Wehrle. 2020. Privacy-Preserving Production Process Parameter Exchange. In *36th Annual Computer Security Applications Conference.*

[74] Christopher Petersen, Ryan J. Caverly, Sean Phillips, and Avishai Weiss. 2023. Safe and Constrained Rendezvous, Proximity Operations, and Docking. In *ACC.*

[75] Christopher D Petersen, K Hobbs, K Lang, and S Phillips. 2021. Challenge Problem: Assured Satellite Proximity Operations. In *AAS SFM.*

[76] Audrey B. Poore, Jeff M. Aristoff, Joshua T. Horwood, Roberto Armellin, William T. Cerven, Yang Cheng, Christopher M. Cox, Richard S. Erwin, Matt D. Hejduk Josehp H. Frisbee, Brandon A. Jones, Pierluigi Di Lizia, Daniel J Scheeres, David A. Vallado, and Ryan M. Weisman. 2016. *Covariance and Uncertainty Realism in Space Surveillance and Tracking.* Technical Report. Numerica Corporation.

[77] Qingyu Qu, Kexin Liu, Wei Wang, and Jinhu Lü. 2022. Spacecraft Proximity Maneuvering and Rendezvous With Collision Avoidance Based on Reinforcement Learning. *IEEE Trans. Aerospace Electron. Systems* (2022), 5823–5834.

[78] Farrukh Bin Rashid, Windhya H. Rankothge, Somayeh Sadeghi, Hesamodin Mohammadian, and Ali A. Ghorbani. 2024. Privacy-Preserving for Images in Satellite Communications: A Comprehensive Review of Chaos-Based Encryption. *arXiv preprint arXiv:2410.21177* (2024).

[79] Rebecca Reesman and Andrew Rogers. 2018. *Getting in your Space : Learning from Past Rendezvous and Proximity Operations.* Technical Report.

[80] Sylvain Renevey and David A Spencer. 2019. Establishment and control of spacecraft formations using artificial potential functions. *Acta Astronautica* (2019).

[81] Hanspeter Schaub and John L Junkins. 2003. *Analytical mechanics of Space Systems.* AIAA.

[82] Katherine Schauer. 2021. *Laser Communications: Empowering More Data Than Ever Before.* https://www.nasa.gov/technology/laser-communications-empowering-more-data-than-ever-before/

[83] Carmine Serio, Guido Masiello, Pietro Mastro, and David C. Tobin. 2020. Characterization of the Observational Covariance Matrix of Hyper-Spectral Infrared Satellite Sensors Directly from Measured Earth Views. *Sensors* (2020).

[84] Adi Shamir. 1979. How to Share a Secret. *Commun. ACM* (1979), 612–613.

[85] Windy S. Slater, Benjamin B.W. Rutherford, Jesse K. Mee, Ryan E. Pinson, Matthew Gruber, Daniel Sabogal, and Ian A. Troxel. 2023. Single Event Effects and Total Ionizing Dose Radiation Testing of NVIDIA Jetson Orin AGX System on Module. In *IEEE Radiation Effects Data Workshop.*

[86] Windy S. Slater, Nayana P. Tiwari, Tyler M. Lovelly, and Jesse K. Mee. 2020. Total Ionizing Dose Radiation Testing of NVIDIA Jetson Nano GPUs. In *HPEC.*

[87] The Aerospace Corporation. 2024. Space Attack Research and Tactic Analysis.

[88] Alyson D. Topper, Jean-Marie Lauenstein, Edward P. Wilcox, Melanie D. Berg, Michael J. Campola, Megan C. Casey, Edward J. Wyrwas, Martha V. O'Bryan, Thomas A. Carstens, Caroline M. Fedele, James D. Forney, Hak S. Kim, Jason M. Osheroff, Anthony M. Phan, Max F. Chaiken, Donna J. Cochran, Jonathan A. Pellish, and Peter J. Majewicz. 2020. NASA Goddard Space Flight Center's Compendium of Radiation Effects Test Results. In *2020 IEEE Radiation Effects Data Workshop (in conjunction with 2020 NSREC).*

[89] U.S. Space Force. 2025. 5 Differences Between the U.S. Space Force and NASA. https://www.spaceforce.com/news-events/5-differences-between-the-us-space-force-and-nasa. Accessed: 2025-11-16.

[90] Zhaokui Wang, Yun Xu, Chao Jiang, and Yulin Zhang. 2019. Self-Organizing Control for Satellite Clusters using Artificial Potential Function in terms of Relative Orbital Elements. *Aerospace Science and Technology* (2019).

[91] Alexandra Weber and Peter Franke. 2024. Space-Domain AI Applications need Rigorous Security Risk Analysis. In *Workshop on Security of Space and Satellite Systems*.
[92] James R. Wertz, David F. Everett, and Jeffrey J. Puschell. 2011. *Space Mission Engineering: The New SMAD*.
[93] Johannes Willbold, Moritz Schloegel, Manuel Vögele, Maximilian Gerhardt, Thorsten Holz, and Ali Abbasi. 2023. Space Odyssey: An Experimental Software Security Analysis of Satellites. In *IEEE Symposium on Security and Privacy*.
[94] John M. Willis, Robert F. Mills, Logan O. Mailloux, and Scott R. Graham. 2017. Considerations for Secure and Resilient Satellite Architectures. In *CyCon US*.
[95] David Woffinden and David Geller. 2007. Navigating the Road to Autonomous Orbital Rendezvous. *Journal of Spacecraft and Rockets* (2007), 898–909.
[96] Yang Xiao, Tao Jiang, Guo-Wei Fan, Liu Zhang, Yu Gao, and Le Zhang. 2024. A meticulous covariance adaptive Kalman filter for satellite attitude estimation. *Measurement Science and Technology* (2024), 045104.
[97] Shu-Yan Xu, Lu-Xiang Xu, Lin-Xiao Cong, Yong-Gui Li, and Cong-Feng Qiao. 2021. First result of orbit verification of Taiji-1 hall micro thruster. *Int. J. Mod. Phys. A* (2021).
[98] Zhihui Xue, Jinguo Liu, Chenchen Wu, and Yuchuang Tong. 2021. Review of In-Space Assembly Technologies. *Chinese Journal of Aeronautics* (2021).
[99] Bruce Yost and Sasha Weston. 2023. *State-of-the-Art of Small Spacecraft Technology*. Technical Report. https://www.nasa.gov/smallsat-institute/sst-soa/
[100] Richard Zappulla, Hyeongjun Park, Josep Virgili-Llop, and Marcello Romano. 2019. Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach. *IEEE Transactions on Control Systems Technology* (2019).

A Communications Capabilities

Satellite communications are necessary for proper spacecraft operation and are comprised of three functions: receiving commands from Earth (uplink), transmitting data and telemetry to Earth (downlink), and relaying information between satellites (crosslink) [99]. Crosslinks, or inter-satellite links (ISL) are increasingly common as the industry moves towards large constellations of satellites [11]. The systems for all satellite communications practices are radio frequency (RF) and free space optical (FSO) or laser communications. Higher frequency bands are generally more desirable as they allow for higher data transmission rates and greater bandwidth. RF band allocations for small satellites in LEO are mainly UHF, S, X, and Ka-bands [99]. More satellites have begun to move toward FSO systems since laser frequencies are much higher than RF (closer to the infrared side of the radio frequency EM spectrum) [99]. Laser communication speeds up data transmission rates by 10–100 times the RF rates [82].

B MP-SPDZ Computation Notes

One discrepancy we noticed in our experiments is that MP-SPDZ sometimes reported a larger number of rounds in the semi-honest model than the number of rounds for the same protocol in the malicious model. This is not expected, as the malicious model computation is strictly larger (and no fewer rounds) than the computation of the corresponding semi-honest protocol. The discrepancy was due to differences in communication for elementary multiplication gates. Parties can send their messages to a dedicated party (king), who then broadcasts the result to others to achieve communication linear to the number of parties. Alternatively, the parties can communicate directly at the cost of asymptotically higher communication, but in a single round. In a three-party setting, the latter is better in terms of concrete costs. We determined that direct communication was not enabled by default for semi-honest Shamir secret sharing, but it was for the malicious model. We thus enabled this in the semi-honest model using the corresponding flag to reconcile the discrepancy. Replicated secret sharing is implemented only for three parties, and the direct flag has no impact on the corresponding protocols. Enabling direct communication (with three parties) does not result in better performance for these protocols, and we evaluate both variants.