2022

Secure Multi-Robot Adaptive Information Sampling with Continuous, Periodic and Opportunistic Connectivity

Tamim Khatib
University of North Florida, khatib.tamim@hotmail.com

Suggested Citation

Khatib, Tamim, "Secure Multi-Robot Adaptive Information Sampling with Continuous, Periodic and Opportunistic Connectivity" (2022). *UNF Graduate Theses and Dissertations*. 1163.
https://digitalcommons.unf.edu/etd/1163

Secure Multi-Robot Adaptive Information Sampling with Continuous, Periodic, and Opportunistic Connectivity

by

Tamim Khatib

A thesis submitted to the
School of Computing
in partial fulfillment of the requirements for the degree of

Master of Science in Computer and Information Sciences

UNIVERSITY OF NORTH FLORIDA
SCHOOL OF COMPUTING
December 2022

The thesis "Secure Multi-Robot Adaptive Information Sampling with Continuous, Periodic, and Opportunistic Connectivity" submitted by Tamim Khatib in partial fulfillment of the requirements for the degree of Master of Science in Computer and Information Sciences has been

Approved by the thesis committee: Date: 12/20/2022

Ayan Dutta.

Ayan Dutta, Ph.D.
Thesis Advisor and Committee Chairperson

Swapnoneel Roy

Swapnoneel Roy, Ph.D.

O. Patrick Kreidl, Ph.D.

Table of Contents

List of Figures

Abstract

Multi-robot teams are an increasingly popular approach for information gathering in large geographic areas, with applications in precision agriculture, natural disaster aftermath surveying, and pollution tracking. In a coordinated multi-robot information sampling scenario, robots share their collected information amongst one another to form better predictions. These robot teams are often assembled from untrusted devices, making the verification of the integrity of the collected samples an important challenge. Furthermore, such robots often operate under conditions of continuous, periodic, or opportunistic connectivity and are limited in their energy budget and computational power. In this thesis, we study how to secure the information being shared in a multi-robot network against integrity attacks and the cost of integrating such techniques. We propose a blockchain-based information sharing protocol that allows robots to reject fake data injection by a malicious entity. However, optimal information sampling is a resource-intensive technique, as are the popular blockchain-based consensus protocols. Therefore, we also study its impact on the execution time of the sampling algorithm, which affects the energy spent. We propose algorithms that build on blockchain technology to address the data integrity problem, but also take into account the limitations of the robots' resources and communication. We evaluate the proposed algorithms along the perspective of the trade-offs between data integrity, model accuracy, and time consumption under continuous, periodic, and opportunistic connectivity.

Chapter 1

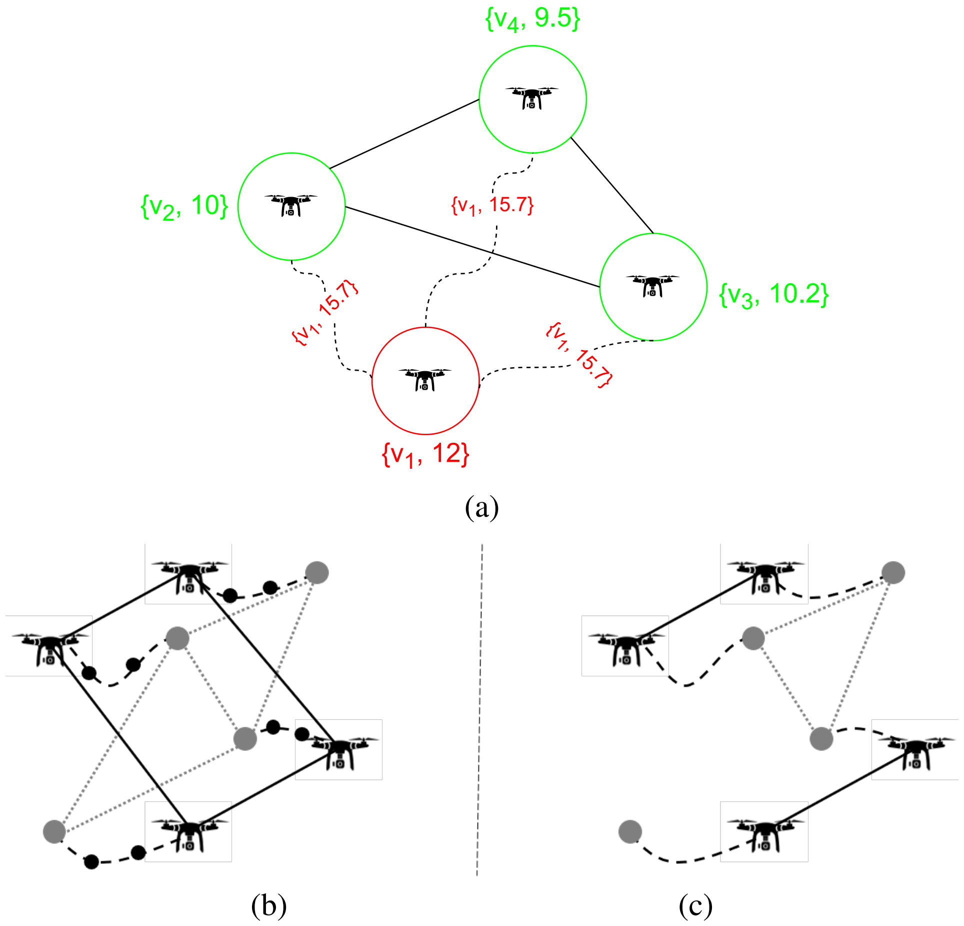
Introduction

In today's era of automation, mobile robots are being deployed for collecting meaningful information from an environment. This has high practical relevance in precision agriculture, search and rescue, and monitoring, among other situations [12, 15, 38]. Such information collection helps human users to make more informed decisions and actions. In particular, with the increased information demand of precision agriculture, aerial or ground robots that collect information about the state of a crop are quickly becoming a standard part of the toolkit of modern farmers [27]. A single robot usually does not have enough capabilities to complete all the relevant tasks and, therefore, multiple low-cost robots are used. As these robots collect and transmit mission critical information to the operation of the farm, the integrity of the collected information becomes a critical concern. Similar to other agricultural machinery, the usage of such robots fluctuates over time. It is thus likely that at any given moment, a farmer might deploy a fleet of robots that are a mix of owned, rented, and borrowed. With such a mix of robots with different provenances, the trustworthiness of individual robots cannot be guaranteed through physical means. Additionally, the multi-robot setting typically assumes the observed data is shared among the robots, which is vulnerable to cyber-attacks. Such attacks can have significant financial and ecological impact [14]. In precision agriculture, farmers use the robots' collected data to decide where to spray herbicides in the field to kill weeds. If a malicious entity breaches the integrity of the collected data, the farmers could unintentionally spray herbicides on the crops rather than the unwanted vegetation.

In this thesis, we study such a multi-robot coordination problem, namely multi-robot information sampling, where the objective is the following: given n mobile robots and a budget B, plan n B-length paths for the robots such that the collected information is maximized [10, 11, 26, 34, 40]. Each robot is equipped with an information collection sensor (e.g., camera, soil salinity meters) and senses information along its path. The information about the environment is represented in the form of a scalar field, while the knowledge model built by the robots is implemented through a Gaussian Process (GP) [29] that integrates all trusted observations made by the robots. The plan is adaptive in the sense that past observations made by the robot(s) affect the future decision-making about where to collect more information. In a multi-robot system, a single robot's future planning is not only affected by its own past sensed data, but also by the observations made by the other robots. To plan such optimal paths is proven to be NP-Hard and, therefore, greedy heuristics for navigation are popularly employed [10, 15, 20, 37]. The problem of multi-robot information collection is under active study but, to the best of our knowledge, how to formally maintain the integrity of the collected information against adversarial influence (see Fig. 1.1) has not yet been studied.

To this end, we propose a blockchain-based secure multi-robot information sampling framework that is resilient against such data integrity attacks. We employ the popular blockchain consensus protocol Proof-of-Work (PoW) to help the robots make decisions based only on the untampered data [6, 13, 17, 23]. Tampered data is detected using the aforementioned consensus protocol and removed from the database. However, integrating such a blockchain-based security protocol comes at a cost – PoW is known to be resource-intensive and will drain the robots' on-board power sources. We thus also analyze the additional energy consumption implied by secure sampling, assessing the cost of our achieved resilience.

Additionally, algorithms used for cryptocurrencies do not directly apply to teams of robots. Because the paths of the robots depend on the collected data (for instance, multiple robots need to converge to explore an area with a disease outbreak), validation of the data needs to be done in real time (e.g., while the drones operate in the air). This creates new challenges with regards to computing capacity and energy usage. Moreover, also in contrast to cryptocurrencies, the connectivity between the nodes might be periodic (i.e., nodes communicate on a shared network only during scheduled times, separated by unconnected autonomous operation) or opportunistic (i.e., nodes communicate only when they fall within range of each other in the course of movement), along with being continuous [1]. For information collection, we employ a popular greedy strategy that is known to yield theoretically-bounded performance [5, 10, 37]. However, it is worth noting that our proposed security technique is generic in nature and can easily be integrated with more sophisticated algorithmic sampling approaches. We tested our proposed framework in a simulation with up to ten robots and benchmarked the results against an insecure baseline, where data integrity attacks are not prevented. This thesis considers scenarios in which the robots have access to continuous, periodic, or opportunistic connectivity and performances of the proposed algorithms are investigated on a variety of scenarios with single or multiple attackers. An illustration is shown in Fig. 1.1.



(a)

(b)

(c)

Figure 1.1. (a) An instance of a data integrity attack studied using blockchain-based security techniques under continuous connectivity (CC) assumptions; an assumed malicious robot (red circled) may send tampered data to its neighbor robots (green circled) in order to degrade future estimates of the underlying information field. We also consider the problem when communication resources are more constrained, specifically under the scenarios of (b) periodic connectivity (PC), in which robots share measurements only in a subset of the rounds (e.g., every three iterations), and (c) opportunistic connectivity (OC), in which robots share measurements possibly only among a subset of the nodes (e.g., those within a range of 4 units).

In summary, the main contributions of this thesis are:

- To the best of our knowledge, this is the first work that studies the secure multi-robot information sampling problem with continuous, periodic, and opportunistic connectivity, which is significant due to its sheer practical relevance.
- To the best of our knowledge, this is the first integration of information sampling and blockchain-based security techniques, specifically the consensus protocol PoW, and studying via simulations the benefit and cost of each.
- We propose and validate practical data integrity algorithms based on the blockchain that are specifically designed to be deployable on robots.
- We study the algorithms along the novel perspective of the trade-offs between data integrity, energy consumption and model estimate error.

Chapter 2

Background

Information gathering using a single or multiple robots has received considerable attention in recent literature [4, 9, 10, 11, 22, 24, 38]. In an informative path planning setting, the objective of the robot(s) is to plan a maximally informative path within a given path-length budget from a start to a goal location, where the robots can be retrieved by human operators [16, 26, 34, 40]. [16] introduced rapidly exploring information gathering (RIG) algorithms that combine ideas from sampling-based motion planning with branch and bound techniques to achieve efficient information gathering in continuous space with motion constraints, while Wei and Zheng [40] proposed a novel informative path planning (IPP) algorithm using reinforcement learning. In particular, Singh et al. [34] developed an efficient Single-robot Informative Path planning (eSIP), which is an approximation algorithm for efficient planning of informative paths that near-optimally solves the NP-hard problem of maximizing the collected information with an upper bound on path-cost.

On the other hand, in a lifelong learning and sampling scenario, such as studied in this thesis, the robot(s) would be given a budget for a particular day's mission and the objective is to collect the maximum information possible within the daily budget [9, 24, 30]. On this topic, [30] introduced the ideas of multi-agent learning and mean field reinforcement learning for multi-robot informative path planning, while integrating recurrent neural networks with mean-field reinforcement learning. Banfi et al. [4] developed two novel asynchronous strategies that work with arbitrary communication models. Viseras et al. [38] have proposed created a non-myopic multi-robot cooperation algorithm for information gathering that could handle the motion constraints of robots, as well as team constraints like communication restrictions. On the other hand, assuming that the obstacles in the environment are not known *a priori*, Dutta et al. [9] proposed continuous region partitioning of the environment into Voronoi components to improve load balancing between robots.

In this thesis, we employ a myopic greedy entropy maximization technique for sampling, which has been shown in the literature to be efficient [5, 10, 20, 37]. Cao, Low, and Dolan [5] presented two approaches to efficient information-theoretic path planning that address a trade-off between active sensing performance and time efficiency. Krause, Singh, and Guestrin [20] proved that the exact optimization of mutual information is NP-complete, and provided a polynomial-time approximation algorithm that is within $(1 - 1/e)$ of the maximum mutual information configuration (the optimum). These information gathering techniques often use a Gaussian Process regressor to model the underlying ambient phenomena, and an information-theoretic metric such as Entropy or Mutual Information is used to drive the robot(s) to meaningful locations where the information gain is maximized [25, 34, 35]. Ma, Liu, and Sukhatme [25] tested an informative planning and online learning method that allowed an autonomous marine vehicle to effectively perform persistent ocean monitoring tasks with a framework that iterated between a planning component and a sparse Gaussian Process learning component. Singh, Krause, and Kaiser [35] presented a non-myopic algorithm, called NAIVE, for informative path planning using multiple robots.

As different parts of the environment might contain significantly different properties of the same ambient phenomena, it is often a good idea to deploy a multi-robot system across disjoint sub-regions in the environment [11, 19, 24]. In [11], the authors proposed a decentralized MDP-based online coordination mechanism that allows a robot to collect maximal information even under control uncertainty. The authors in [19] investigated a multi-robot coordination approach for informative sampling with autonomous underwater vehicles, while [24] presented an adaptive sampling algorithm for learning the density function in multi-robot sensor coverage using Mixture of Gaussian Processes models. Researchers have only begun to look into reinforcement learning-based information sampling technique while using the Gaussian Process as an information modeling tool. One of the first such study with a multi-robot system is due to Said et al. [30], where a deep Q-learning technique has been used with the help of a recurrent neural network.

As a robot's future decision-making, as well as path planning, depend on the previously collected data (locally and communicated by others), tampered data can create havoc. The preservation of data integrity via a blockchain-based solution using the Proof-of-Work (PoW) consensus protocol was examined in [36], each assuming the robots enjoyed continuous connectivity (CC). PoW is a popular and effective choice in the cryptocurrency industry [28], but it is known to be resource intensive [7, 8] and, in turn, raises new challenges for resource-limited multi-robot information collection. For example, De Vries [7] noted that the Bitcoin network, which uses PoW as its consensus protocol, consumed at least 2.55 gigawatts of electricity in 2018 and could potentially consume 7.67 gigawatts in the future, making it comparable to countries such as Ireland (3.1 gigawatts) and Austria (8.2 gigawatts).

Most of the multi-robot information collection techniques in the literature assume the communication among the robots is always available, and therefore, the coordination among them is continuous. In this thesis, we follow this principle by first assuming continuous connectivity among the robots as illustrated in Fig. 1.1. Exploration while maintaining such connectivity under limited communication ranges is studied in [10], using a graph-theoretic technique. For conflict-free, multi-robot informative path planning, the authors in [26] used a bipartite matching-based technique while adapting it to handle spatio-temporal dynamics. While CC is the most common assumption in literature when it comes to multi-robot information collection techniques, it is not the most realistic one. There are many scenarios where the robots will not necessarily be in constant contact. Additionally, because PoW is incredibly resource-intensive, CC may only worsen the performance issues. On the other hand, if the robots were to connect only periodically (PC), the optimal multi-robot re-connection planning problem has been proved to be NP-hard even when the environment is modeled as a tree [3]. Heuristic solutions are presented in the literature for such settings [3, 15, 22]. Specifically, Hollinger and Singh [15] introduced the concept of multi-robot informative path planning using PC (MIPP-PC), while Lauri, Heinänen, and Frintrop [22] developed the ρ-decentralized partially observable MDP model, which outperformed previous heuristics.

The final connectivity technique that we consider is opportunistic (OC), where the robots are given a limited range for communication, restricting coordination with others to only when another robot is in the vicinity. As this does not pose any connectivity policy restrictions and most closely resembles most real-world situations, this strategy has recently been adopted in general multi-robot exploration studies [2, 9, 11]. How to coordinate a multi-robot system to clear blocked paths was addressed in [2], where a collaborating robot would need to interrupt its current exploration and move to a different location to collaboratively clear a blocked path. A survey and analysis of various connectivity models for multi-robot exploration and coordination can be found in [1]. However, none of the aforementioned works consider adversarial influence, such as tampering with measured data, on multi-robot information collection.

Chapter 3

Problem Setup

We have a set of n robots $R = r_1, r_2, \cdots r_n$. The robots are homogeneous, localized using a GPS, and move in a shared environment. The environment is discretized into a graph $G_p = \{V, E\}$, where the node set V represents the information collection locations and the connections among them are denoted by the edge set E. Each robot r_i has its unique subregion for exploration, $\mathcal{V}_i$, and $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$. $\mathcal{V}_i$ can be calculated in a pre-processing stage by applying Voronoi partitioning [39] or K-medoids clustering [18]. Without loss of generality, we assume that $\cup_{i=1}^{n} \mathcal{V}_i = V$. The action set of the robots is denoted by A. For example, in a 8-connected grid G_p, A will hold the motor commands to move to all the eight neighbors. r_i is equipped with an on-board sensor using which it can sense and collect information (e.g., camera). The robots' observations are modeled to be noisy. A robot r_i starts from a node $v_i^0 \in \mathcal{V}_i$. The robots are sensing an ambient phenomenon $\mathcal{Z}$ that varies with the location, with $\mathcal{Z}(v_i^0)$ being the (scalar real) value at node v_i^0.

With CC, we assume that a robot r_i can communicate with $r_j, \forall r_j \in R \setminus r_i$ after collecting data at any node, i.e., the robots maintain a continuous connectivity throughout the exploration. The algorithmic details of maintaining such a network is out of scope for this work; an example of techniques that can be used are the ones proposed in [10]. In PC, the robots will form a connected network after every $\mathcal{F}$ cycles named coordination frequency [15, 22]. The reader is referred to [3, 15] to see how such reconnections can be established periodically. In OC, the robots are not guaranteed to form connected communication networks, instead communicating if and when two or more robots are within each other's communication ranges (C).

One should note that with OC, one robot might never communicate with another robot during the exploration, and it is also possible that the robots form disconnected sub-networks [9, 11].

We use a Gaussian Process (GP) to model the uncertain environment. Let $\mathbf{X}$ denote a Gaussian random vector of length $|V|$ with prior mean vector μ and covariance matrix Σ, where μ and Σ represent the prediction in node set V and its corresponding uncertainty, respectively [29]. The volumetric measure of this uncertainty is calculated by an information theoretic metric, (differential) entropy, which is formally defined as

$$H(\mathbf{X}) = \frac{1}{2}\log|\Sigma| + \frac{|\mathcal{V}|}{2}\log(2\pi e). \tag{3.1}$$

Each robot starts with a common initial GP model, GP^0, and then takes measurement $\mathcal{Z}(v_i^0)$ at the start node $v_i^0 \in V$. We assume the measurements are subject to additive white Gaussian noise $\epsilon \in \mathcal{N}(0, \sigma)$. The updated local GP, GP_i, for robot r_i is then given by the posterior statistics:

$$\begin{aligned} \Sigma_i =& \Sigma - \Sigma\mathbf{C}\left(v_i^0\right)' \left(\mathbf{C}\left(v_i^0\right)\Sigma\mathbf{C}\left(v_i^0\right)' + \right. \\ & \left. \sigma_n^2\right)^{-1}\mathbf{C}\left(v_i^0\right)\Sigma \\ \mu_i =& \mu + \Sigma_i^0\mathbf{C}\left(v_i^0\right)' \left(\mathcal{Z}(v_i^0) - \mathbf{C}\left(v_i^0\right)\mu\right)/\epsilon, \end{aligned} \tag{3.2}$$

where $\mathbf{C}\left(v_i^0\right)$ denotes the length-$|V|$ row vector of all zeros except for a one in component v_i^0 and $\mathbf{C}\left(v_i^0\right)'$ is its matrix transpose. The reader is referred to [11] for more details.

It is a standard assumption in kernel-based parametrizations of GPs that the correlation between two nodes are inversely proportional to the distances between them [11, 20, 29]. We exploit this property when computing entropy by approximating the computationally intensive matrix determinant $|\Sigma|$ by the product of the per-node variances (σ_v^2) along the diagonal of Σ. In turn, the associated entropy $H(\mathbf{X})$ decomposes additively across the nodes, each per-node term $(H(X_v))$ given by

$$H(X_v) = \frac{1}{2}\log\left(2\pi e\sigma_v^2\right). \tag{3.3}$$

Our proposed techniques utilize these per-node entropies, their sum (via the Hadamard inequality) serving as upper bound for the true global entropy $H(\mathbf{X})$, to drive the robots to opportune locations for information collection. Each robot's local GP model, GP_i, is initialized with a training dataset $\mathcal{D}$, and the prior statistics are calculated before it is deployed in node $v_i^0 \in \mathcal{V}_i$. After deployment, each robot first collects the information in v_i^0 and this observed data is used along with $\mathcal{D}$ to calculate the per-node rewards using using Eq. 3.3. In a greedy fashion, r_i then chooses the next adjacent node $v_i^* \in \mathcal{V}_i$ that provides the maximum information,

$$v_i^* = \arg \max_{v \in adj(v_i^0)} H(v|\mathcal{D} \cup \mathcal{Z}(v_i^0)), \text{ s.t. } v \in \mathcal{V}_i. \tag{3.4}$$

In the absence of communication, each robot will continue this $sense - and - move$ cycle until it runs out the given budget B. Such greedy strategies have been observed to be efficient in the literature, and in certain conditions (albeit not being satisfied here) even provably so [5, 20, 34].

Chapter 4

Algorithms

It is not the objective of this thesis to develop a new algorithm for information sampling; rather, we study how an integrity-preserving blockchain-based protocol can be integrated with the information collection framework presented in [1, 10, 37]. This thesis is interested in studying the resilience against data integrity attacks within the constrained communication setting, specifically under continuous (CC), periodic (PC) and opportunistic connectivity (OC).

4.1 PoW Consensus Protocol in CC

In the absence of a security protocol, each robot takes the received information from the other robots into account and updates the local GP model using Eq. 3.2 (see Algo. 1). One or more malicious entities can attack this data sharing system via data tampering attempts and denial-of-service (DoS) [14, 21].

To prevent other robots from incorporating such fake data in their future decision-making, we have used a blockchain-based security protocol. Blockchains are tamper-resistant digital ledgers that the robots maintain in a distributed fashion [31]. In a blockchain, the data is stored in discrete units, called blocks, that are linked (chained) to each other by having the hash of one block be part of the data of the next block. Each robot r_i maintains a local blockchain $\mathcal{C}_i$. Each block $b_{idx} \in \mathcal{C}_i$ contains the following components $< D, T, idx, \mathbf{N}, H_{last} >$, where D denotes the collected measurement(s), T represents the current timestamp, idx is the index of the block, $\mathbf{N}$ is an integer called nonce, and finally, H_{last} represents the previous block b_{idx-1}'s hash. We particularly use blockchains because of their chain data structure – if an attacker is

Algorithm 1: Secure Information Sampling with PC and OC

```
/* Every robot follows a <move-sense-communicate-estimate> cycle */
$r_i$ calculates the next best location $v_i^*$ to move to;
while budget left do
    Move to $v_i^*$ and Sense information $\mathcal{Z}(v_i^*)$;
    Create a block $b_{idx}$ that includes $v_i^*$ and $\mathcal{Z}(v_i^*)$;
    Add $b_{idx}$ to $\mathcal{C}_i$ and broadcast it 1) every cycle with OC, or 2) periodically every $\mathcal{F}$ cycle
      with PC;
    $\tilde{\mathcal{C}} \leftarrow$ receive similar blockchains from 1) $\forall r_j \in R \setminus r_i$ with PC, or 2) $\forall r_j \in \bar{R}$ with OC;
    Secure. Decide to add the measurements from $\tilde{\mathcal{C}}$ to $\mathcal{C}_i$ or not using PoW;
    Estimate. update $GP_i$ with the new data in $\mathcal{C}_i$ (Eq. 3.2) and update the entropies (Eq. 3.3);
    Select $v_i^*$ based on the updated entropies (Eq. 3.4);
```

able to modify D in block b_x, the hash of the block will also change, and therefore, it will then not match H_{last} in b_{x+1}.

After r_i measures $\mathcal{Z}(v_i^*)$ at v_i^*, it puts them in D. The nonce is initially set to zero. The robot creates a block with it and finds its corresponding hash. To mine this block, r_i checks whether the hash has the required *difficulty* or not. The difficulty of a block is represented by the leading zeros in the hash – the higher the number of zeros in the beginning of the hash, the more difficult it is to mine. We use an iterative nonce setting approach, i.e., if the nonce does not produce a hash with the desired difficulty, we increase the nonce by one. This process continues until the desired nonce, and more importantly, the desired difficulty in the corresponding hash is found. Once this mining process is over, the block is placed into r_i's local blockchain $\mathcal{C}_i$. With CC, the robots share their newly created blocks among each other after every cycle of sense and measurement. The robots replace their local blockchains with the received blockchains if the blocks are validated, and as a result, at the end of each coordination cycle, every robot will have other robots' valid new blocks along with their existing blocks in their local blockchains (see Algo. 2). Note that the verification of the hash is straightforward. A robot looks at the nonce in a particular block, finds its corresponding hash, and checks whether the hash has the desired difficulty level. If not, the block is rejected; otherwise, it is validated. Naturally, increasing the difficulty reduces the probability of it being compromised while the time and energy required by the robots increase significantly.

Algorithm 2: Multi-robot Proof-of-Work (PoW) Consensus Protocol

```
Input: $\mathcal{C}_i \leftarrow r_i$'s local blockchain;
       $\mathcal{C}_j^r \leftarrow$ received blockchain from $r_j$;
if $len(\mathcal{C}_j^r) > len(\mathcal{C}_i)$ AND checkChainValidity() then
    $\mathcal{C}_i \leftarrow \mathcal{C}_j^r$;
Procedure checkChainValidity()
    $H(b_{idx}) \leftarrow$ hash of block $b_{idx}$;
    $H_{last}(b_{idx}) \leftarrow$ hash of the previous block $b_{idx-1}$ stored in $b_{idx}$;
    for each block $b \in \mathcal{C}_j^r$ do
        if isValidBlock($b_{idx}$) is false OR $H_{last}(b_{idx}) \neq H(b_{idx-1})$ then
            return false
    return true
Procedure isValidBlock()
    $d(b_{idx}) \leftarrow$ difficulty of block;
    $d_{min} \leftarrow$ minimum valid difficulty;
    if $d(b_{idx}) \geq d_{min}$ then
        return true
    else
        return false
```

4.2 PoW Consensus Protocol in PC and OC

With PC, r_i creates D with the last $\mathcal{F}$ collected measurements. As the robots coordinate periodically, they do not get a chance to share their collected information every cycle. Therefore, each block will contain $\mathcal{F}$ measurements in PC whereas it contained only one in CC. The other components in the block are calculated in the same way as in CC. Having a larger block size has one advantage – the robots do not need to share information in every cycle, and therefore, the communication and mining overheads are significantly less. On the other hand, in a bandwidth-limited environment, sharing a larger block might be prohibitive. Furthermore, as the robots are not aware of others' collected data, the quality of their informative paths might be sub-par compared to CC.

With OC, when two or more robots $\bar{R} \subseteq R$ come within each other's communication ranges, they share their local blockchains and the coordination happens in the same way as in CC. Each robot $r_i \in \bar{R}$'s local blockchain contains its observed data and any valid data it has received earlier from $r_j \in R$. As the robots are collecting data from disjoint sub-regions in the environment, they might have mutually exclusive local blockchains. This might lead to *orphan* blocks.

An orphan block is a block that was mined and placed in the blockchain at some point. However, over time, a new blockchain was generated that did not include this block, leaving it abandoned. Orphan blocks only exist in OC. For example, suppose robot r_i has a local blockchain containing the following blocks $\{a, b, c, d\}$, and Robot r_j has a local blockchain of $\{a, b, c, e, f, g\}$. Next, these two robots come within C distance. Following our algorithm, r_i will accept the longer blockchain of r_j, causing block d to be abandoned, namely an *orphan block*. While Block d in particular will no longer be used, the data within it will be extracted and put back into a memory buffer known as *unconfirmed data* that r_i maintains in OC for such scenarios.

Note that this is *not* the same as block d; the data D in it is the same, but the previous hash, the timestamp, and the nonce will all be different. Also, block d was still a valid block, but was left out of the blockchain simply due to asynchronous coordination in OC and not because of malicious data. Although the data in block d is preserved, the block itself will stay orphaned, meaning the mining effort put into it is lost.

Lemma 1. *Using our proposed algorithm, the robots will not lose any observed data.*

Proof. Consider a scenario with two robots r_i and r_j and suppose at a particular point in time r_i's blockchain is larger than r_j's blockchain. Additionally, r_j's blockchain contains a particular observed data x. We claim that the observed data x will not be lost after r_i and r_j coordinate. Assume the contrary, which is that data x will be lost. If r_i's blockchain does contain x, then x cannot possibly be lost, because when r_j accepts r_i's blockchain, x will be among the accepted data. On the other hand, if r_i's blockchain does not contain x, r_j will take x and place it back in its unconfirmed data, a data structure that contains data not yet included in the blockchain. Once r_j accepts r_i's blockchain, it will then insert x at the end of the blockchain after mining. If r_j has not already mined this cycle, x will be restored to its blockchain, so x cannot possibly be lost in this case. However, if r_j has already mined this cycle, x will be restored to the blockchain on the next cycle. It follows that it is impossible for x to be permanently lost. □

Chapter 5

Experiments

5.1 Settings

We have implementing the proposed secure information sampling techniques with up to 10 robots in MATLAB and Python. The robots are placed randomly in an 8-connected 14×14 grid environment. The robots can only visit up to 20 nodes within their own sub-regions $\mathcal{V}_i$. We have sampled our underlying ground truth information for 196 grid locations from a zero-mean Gaussian random vector, where the covariance matrix represents an exponential kernel function: specifically, for any pair of nodes v_s and v_t, the covariance between them is represented by

$$\beta^2 \exp\left(-||v_s - v_t||/\ell\right)$$

where hyperparameters $\beta > 0$ is the local standard deviation and ℓ is the exponential rate of diminishing covariance between increasingly distant nodes. In our experiments, β and ℓ are set to 1 and 25, respectively. The additive white Gaussian noise $\epsilon \in \mathcal{N}(0, 0.25)$.

The adversarial influence is modeled as commanding the same subverted node to falsify its measurements only periodically, leaving that node's measurements and behavior unaltered otherwise. More specifically, all experiments with adversarial influence assume one or more nodes are subverted, falsifying measurements once every four rounds. The falsification process itself is also simplistic; specifically, the adversary chooses a magnitude uniformly from the range $[-10, +10]$.

We compare our proposed PoW-based algorithm with CC, PC, and OC assumptions against two benchmarks:

- *No Attack*. In this scenario, there is no malicious robot in the system, and therefore, there is no chance of data tampering;
- *Insecure*. In this scenario, data integrity attacks are similar to the attacks on our algorithms, however, there is no security protocol in place to protect against such attacks.

5.2 Results

First, we are interested in investigating the effect of the security attacks on the quality of the information model estimated by the robots. To analyze the effects of data integrity attacks on multi-robot information sampling, we investigate the mean square error (MSE) metric that represents how close to the ground truth the predicted information model is. The results are presented in Figs. 5.1 and 5.2(a). The shaded regions indicate the standard deviations.

5.2.1 Single Attacker

5.2.1.1 CC MSE

For all of PoW, we have varied the difficulty level between $[1, 5]$, i.e., between one and five leading zeros in the hash. We see that with more robots, the average effect of the data integrity attacks on MSE usually minimizes. As we only have one robot that sends tampered data to the other robots, with $n = 2$, that accounts for 50% robots being malicious, whereas with $n = 10$, it only accounts for 10%. For example, the final MSE for CC with these two values of n are 0.78 and 0.19, respectively.

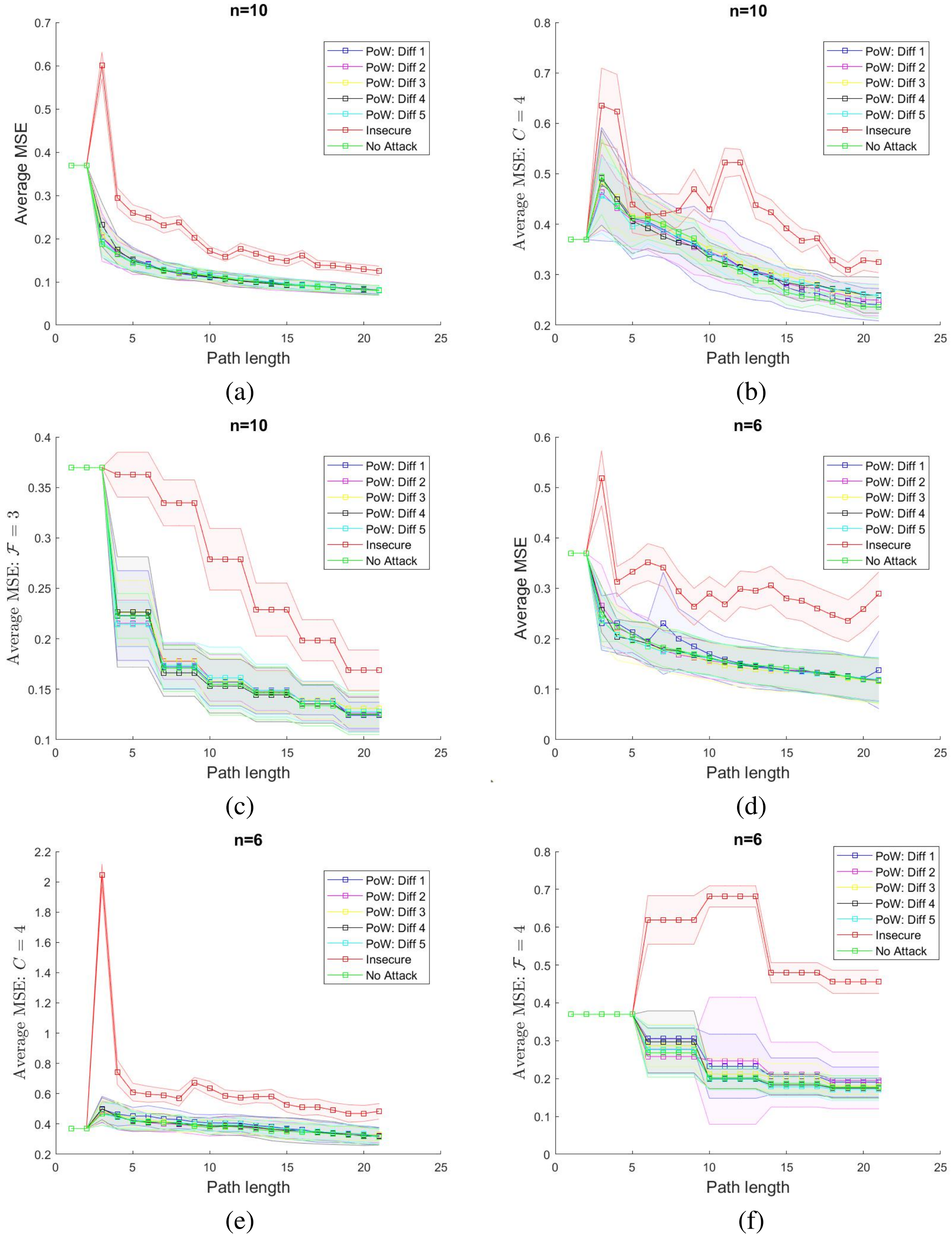
In any case, the MSE decreases as the robots make more observations [5, 10, 20]. However, if we have an insecure system and there is an integrity attack, we see the MSE values jump to higher numbers any time there is an injection of tampered data. As the attack happens periodically, the spikes in the plots are also periodic. We also observe that if the PoW difficulty is low, e.g., 1, the attacker might get 'lucky' due to the relatively high probability of the hash being found and, consequently, the security breaking down. Since there are 16 possible hash values per digit and only one digit is an acceptable value for the prefix (0), the probability that the hash satisfies the difficulty 1 condition is $\frac{1}{16}$. For difficulty 2, this would probability is $(\frac{1}{16})^2 = \frac{1}{256}$; for difficulty 3, $(\frac{1}{16})^3 = \frac{1}{4096}$, and so on. In our experiments, there were a total of 375 attacks with difficulty level 1; 28 of those attacks succeeded in successfully tampering the data and letting other robots use that for estimation, which has a probability of $\frac{28}{375} \approx \frac{1}{13.4}$. This is illustrated by an MSE spike for $n = 6$ in Fig. 5.1(d). However, when we increase the level of the difficulty, e.g., 5, the MSE values coincide with the MSE values in the *No Attack* scenario.

5.2.1.2 PC MSE

When we compare the MSE results for PC against the *Insecure* version, PC almost always performs statistically significantly better. Similar to CC, the blockchain-based proposed technique will occasionally fail to safeguard against the data tampering attempts if the selected difficulty is low, e.g., 1. When we compare the PC results with varying $\mathcal{F}$, the robots performed better, when they communicated more often (e.g., $\mathcal{F} = 2$ is better than $\mathcal{F} = 5$). However, an interesting thing to note is that while this trend was consistent, they rarely resulted in a large difference in MSE. We believe that the small range in MSE results due to various frequencies are because that the robots that coordinate more often have more opportunities to adapt their plans to explore better (see Fig. 5.2(a) for reference). In general, the closer the connectivity model is to CC, the better PC performs. Note that this also results in a higher computation time, which we will discuss later in this section.

5.2.1.3 OC MSE

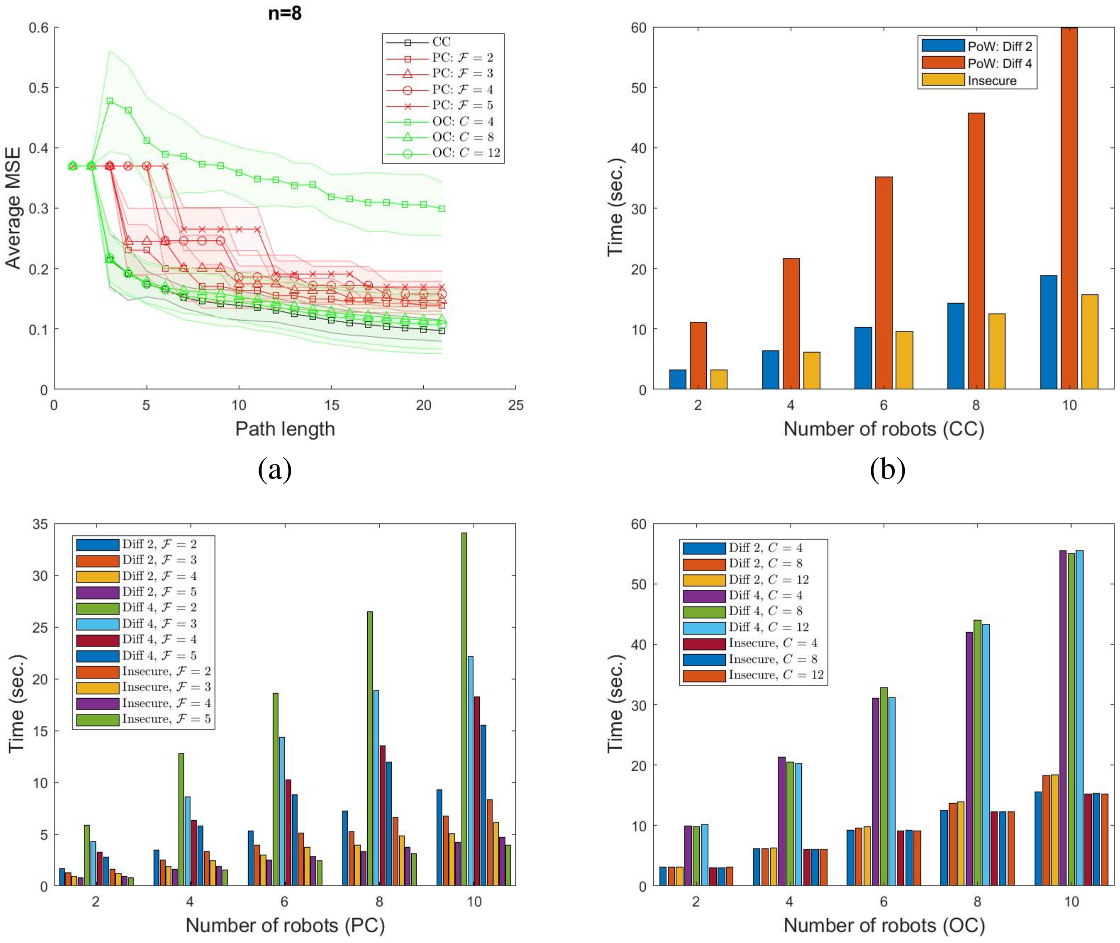
Similar to CC and PC, the OC model almost always performs statistically significantly better than the *Insecure* version (except for a few cases with difficulty 1) due to the reasons discussed earlier. We have found that with a higher C, the MSE is lower than compared to a smaller C. The difference in MSE with various communication ranges is significant. For example, with difficulty 4 and n set to 4, the final MSE value with $C = 4$ is 0.33, whereas with $C = 12$, it is 0.14. In nearly every experiment, $C = 12$ outperformed $C = 4$ by a statistically significant amount. $C = 8$ resembled $C = 12$ when there were 8 and 10 robots, with a small difference for 6 robots and a clear difference at the edge of statistical significance for 4 robots. With 2 robots, the MSEs with $C = 8$ seem more similar to $C = 4$ than $C = 12$. This trend is largely



(a) (b) (c) (d) (e) (f)

Figure 5.1. Single Attacker: MSE comparison (lower the better) among various connectivity models used: (a,d) CC with $n = 10$ and 6; (b,e) OC with $n = 10$ and 6; and (c,f) PC with $n = 10$ and 6.

due to a transitive feature in the communication of our robots. If Robot B can communicate with both Robot A and Robot C, Robot A and Robot C can communicate with one another



(a) (b)

(c) (d)

Figure 5.2. Single Attacker: (a) Comparison of MSE values among all the connectivity models with $n = 8$; Run time comparison (lower the better) between our proposed secure techniques and the implemented benchmark algorithms: (b) CC; (c) PC; and (d) OC.

using Robot B as an intermediary, even if Robot A and Robot C are out-of-range on their own. Due to this property, when there are more robots, communication range matters less since two robots out-of-range can still communicate if there is a third robot in range of the other two. So having a range of 8 instead of 12 made a much larger difference (up to 5.5 times when n increases from 4 to 8 with difficulty 4).

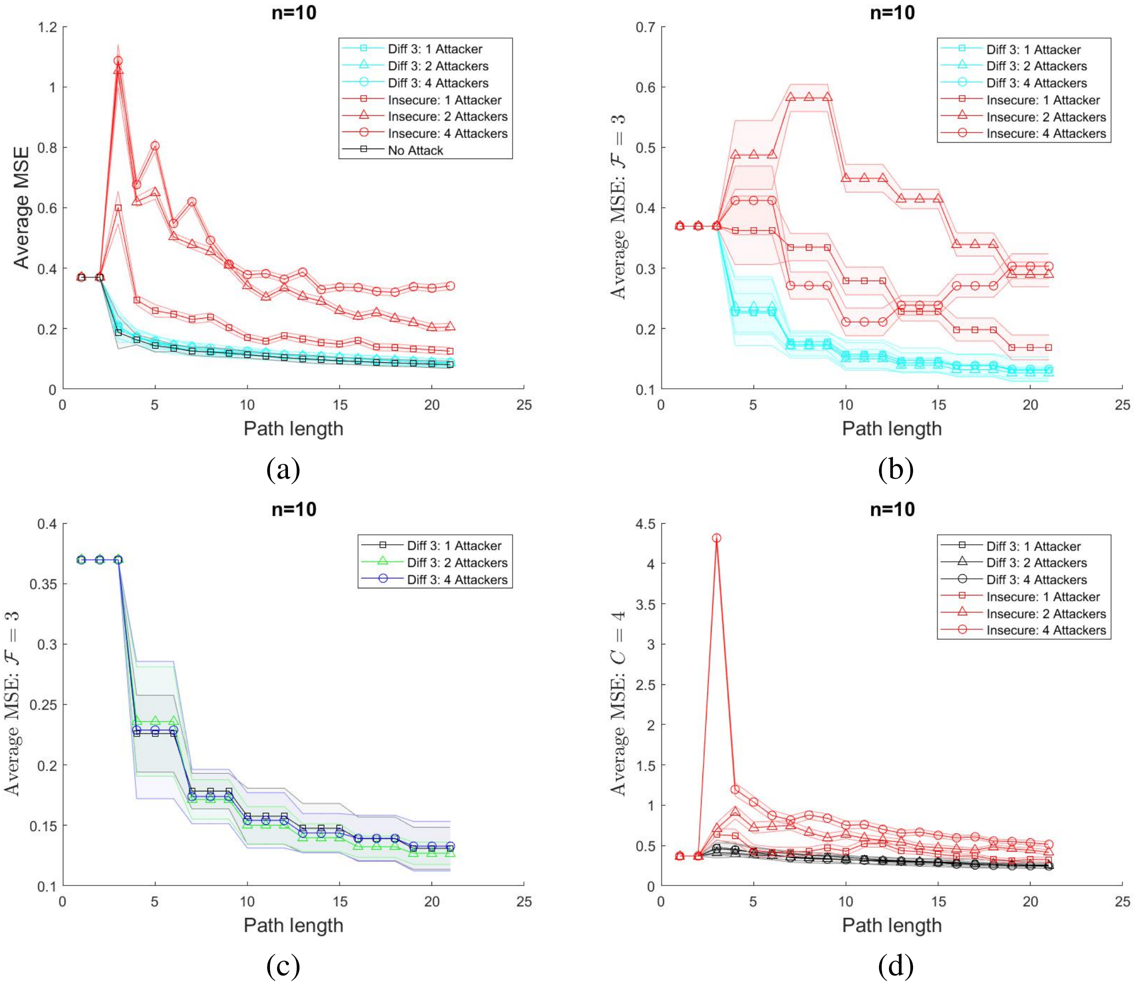
5.2.1.4 Summary MSE

When all three connectivity models are compared together (Fig. 5.2(a)), CC always performed the best. This is expected, as CC is essentially a specialized PC and OC where the robots meet every iteration and have infinite range. Both PC and OC performed significantly worse when faced with more constrained conditions, i.e., reduced meeting frequencies for PC and reduced range for OC. In particular, OC performed the worst among the three connectivity models when there were few robots since that meant they would rarely communicate. PC always performed noticeably worse than CC as the robots with CC always communicate and coordinate, and therefore, the robots could adapt their joint paths on a finer scale. For OC, however, this was not the case. When C and n are high, OC became almost identical to CC due to all the robots sharing their collected data after every round and fine-tuning their paths.

5.2.1.5 Time

As discussed previously, although the PoW consensus protocol allows us to achieve secure information collection, it also consumes a significant amount of resources. To measure such effects, we investigate the run time metric next. We see that with a higher difficulty level in PoW, the run time significantly increases. For example, with 10 robots ($n = 10$) and difficulty 1, the run time is 19.18 seconds, whereas with difficulty 5, the run time is 694.74 seconds. This is due to the fact that each attempt at satisfying a difficulty 1 hash has a $\frac{1}{16}$ chance of succeeding, while satisfying a difficulty 5 hash has a $(\frac{1}{16})^5 = \frac{1}{1{,}048{,}576} \approx 9.54 \times 10^{-7}$ chance, indicating far more attempts must be made. On the other hand, in the *Insecure* scenario, the corresponding run time is 15.02 seconds. We can see that there is an increment of 46.25 times in the run times to protect the information against data tampering attempts.

PC always outperformed CC (Fig. 5.2(b-d)). Furthermore, with PC, the run time is lower in cases when robots coordinate less often. For example, with $n = 10$ and difficulty 4, the run times for PC with $\mathcal{F} = 2$ and 5 are 34.04 and 15.50 seconds, respectively, while the run time for CC is 59.76 seconds. On the other hand, OC always performed better than CC, but worse than PC. Additionally, while OC usually did slightly better when range was greater, in some cases, this was not followed. This is because that there were certain occasions when the time saved from coordinating less often was counterbalanced by the time spent redoing PoW for the orphan blocks.



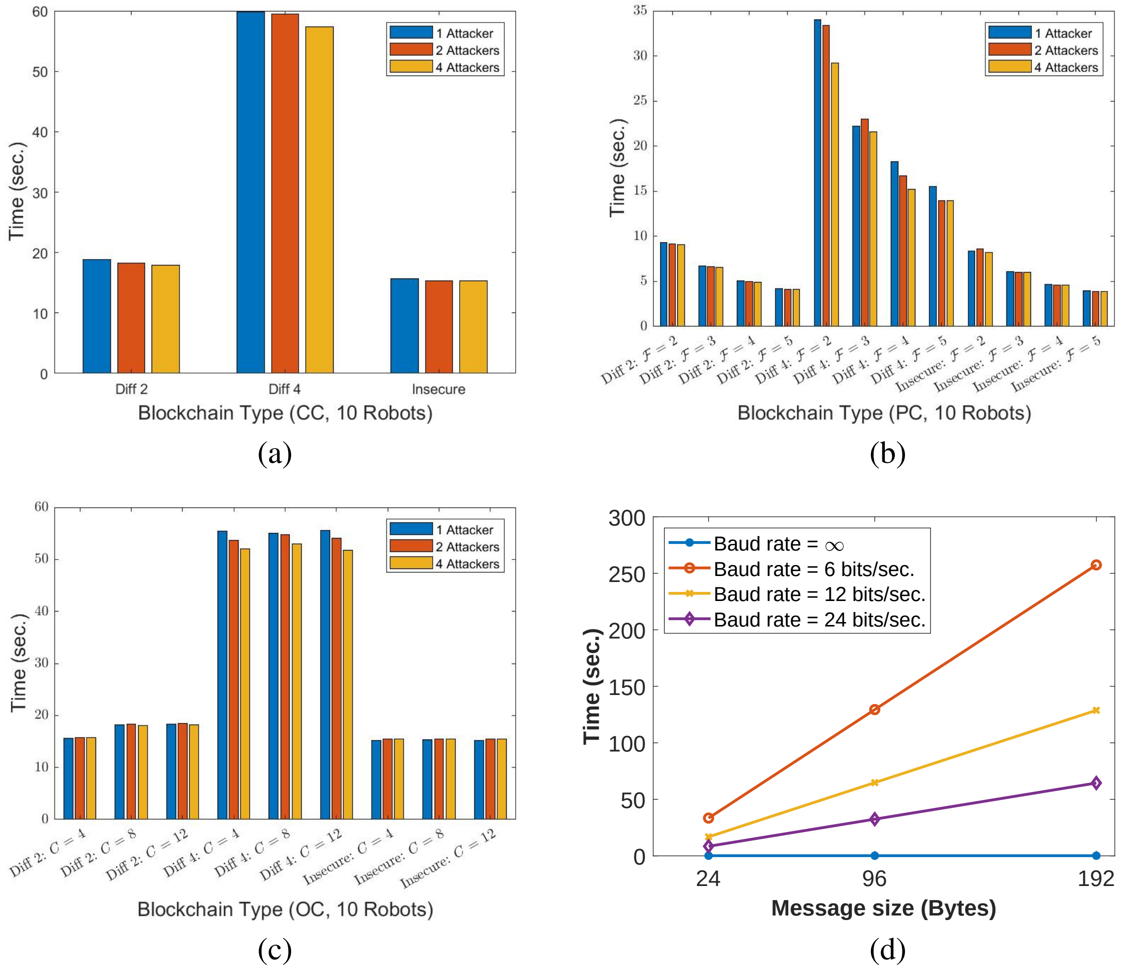
(a)

(b)

(c)

(d)

Figure 5.3. Multiple Attackers: MSE comparison (lower the better) among various connectivity models used: (a) CC; (b) PC; (c) PC zoomed in on our algorithm's results; and (d) OC.



(a) (b)

(c) (d)

Figure 5.4. Run time comparison (lower the better) among various connectivity models used with Multiple Attackers: (a) CC; (b) PC; (c) OC; and (d) Effect of various Baud Rates and message sizes on run time ($n = 2$).

5.2.2 Multiple Attackers

For CC, PC, and OC, the difference in their MSE values with 1, 2, and 4 malicious robots was small (Fig. 5.3). As the objective of PoW is to ensure that attacks do not affect accuracy, though having more attacks did lead to a slightly worse performance in most cases, the difference was insignificant. For example, with one malicious attacking robot, $n = 10$, and $\mathcal{F} = 3$ in PC, the final MSE value is 0.131, whereas with four attacking robots, the MSE value is 0.133. This slight difference is likely because even if the attacker was unable to add fake data to the blockchain, it still deleted all of the compromised robots' unconfirmed data while degrading the information model. In terms of time, more malicious robots in the system led to a small decrease in run time. This likely because that while PoW takes up the majority of the computation time, another time-consuming operation is transferring the data from one blockchain to another. Since this transfer of data occurs less often due to more attacks, run time decreases as fewer uncompromised robots transfer the blockchain data among themselves.

5.2.3 Effect of Baud Rate

Finally, we are interested in investigating how with baud rate – data transmission rate in a communication channel – the communication time changes. For this, we used Webots, a high-fidelity 3D simulator, where one robot is set up to send data and the other is set up to receive it. In CC, coordination happens after every round of data collection, and therefore, a message in CC containing a block is smaller than when compared to PC, where the robots share the past $\mathcal{F}$ collected data in a single message. The result is presented in Fig. 5.4(d). When the baud rate is set to infinity, a standard assumption in multi-robot coordination studies [9, 10, 11, 19, 24, 38], the communication time is almost negligible, the maximum being 0.13 seconds. On the other hand, when it is restricted to be only 6 bits per second, to send a 192-byte message (the equivalent of putting the past eight observations in a message), it takes 257.40 seconds, whereas for a 24-byte message, the communication time is 33.40 seconds. This result is significant in terms of CC, PC, and OC comparisons. Although PC needed a fraction of the computation time of CC, it might not be a good choice in case of a limited-bandwidth environment. This is also partially true for OC as the communication among the robots is not algorithmically determined, meaning the robots may need to exchange large chunks of data if and when they come within each other's communication ranges.

The research work presented in this thesis has been published in [32, 33].

Chapter 6

Conclusion and Future Work

In this thesis, we considered the task of multi-robot adaptive information sampling in a setting where robots are collaborating to obtain a high-quality global model of an ambient phenomenon by updating a Gaussian Process-based estimate. We posit that such systems are vulnerable to adversaries inserting false information into the model due to the dependence of the path planning decisions of the robots on the current model. We proposed a secure multi-robot information sampling algorithm where the robots rely on a blockchain-based technique to accept or reject incoming observations. We proposed and implemented a variation of the blockchain technique based on the Proof-of-Work consensus protocol. We performed an extensive set of experiments assuming threat models with a single or multiple attackers. We found that by varying the number of digits in the hash prefix, we can trade off between the energy consumption and the integrity guarantees of the data. As our setting involves an estimation technique that uses a Gaussian process, the estimation is robust to occasional incorrect observations. Thus, even a hash prefix of a single digit can achieve an acceptable error in the estimate. Experiments show that the algorithm can significantly improve the quality of the information model in the presence of a persistent attacker. However, there is a tradeoff between the number of zeros requested in the hash for the PoW algorithm, and thus implicitly the quality of the model, and the computational cost. We found that the proposed algorithms, while effective against the considered attack, add a significant computational overhead to the robot. These results vary depending on whether the robots operate under CC, PC, and OC connectivity model, with CC having the best accuracy and the worst time, PC having predictable gains and losses in accuracy and time depending on the meeting frequency, and OC having an accuracy that is harder to predict and a time that does not significantly vary when the communication range changes.

Our results suggest several future directions of research. Depending on the particular scenario, it might be possible to determine, in real-time, the optimal choice of the difficulty in the blockchain algorithm for a specific balance of the computational cost and model accuracy. A better understanding of the relationship between the model and the path chosen by the robots could also allow for a partial offloading of the blockchain computations to the cloud after the data sampling had been completed. Other potential future work include the extension of the proposed algorithm to path planning algorithms that react to changes in the environment, and extensions of the proposed approach that further improve scalability.

Acknowledgement

This work was supported in part by the National Science Foundation (NSF) Cyber-Physical Systems Grant #1932300.

Bibliography

[1] AMIGONI, F., BANFI, J., AND BASILICO, N. Multirobot exploration of communication-restricted environments: a survey. *IEEE Intelligent Systems 32*, 6 (2017), 48–57.

[2] ANDRE, T., AND BETTSTETTER, C. Collaboration in multi-robot exploration: to meet or not to meet? *Journal of intelligent & robotic systems 82*, 2 (2016), 325–337.

[3] BANFI, J., BASILICO, N., AND AMIGONI, F. Multirobot reconnection on graphs: Problem, complexity, and algorithms. *IEEE Transactions on Robotics 34*, 5 (2018), 1299–1314.

[4] BANFI, J., LI, A. Q., REKLEITIS, I., AMIGONI, F., AND BASILICO, N. Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Autonomous Robots 42*, 4 (2018), 875–894.

[5] CAO, N., LOW, K. H., AND DOLAN, J. M. Multi-robot informative path planning for active sensing of environmental phenomena: a tale of two algorithms. In *International Conference on Autonomous Agents and Multi-Agent Systems, AAMAS '13, Saint Paul, MN, USA, May 6-10, 2013* (2013), M. L. Gini, O. Shehory, T. Ito, and C. M. Jonker, Eds., IFAAMAS, pp. 7–14.

[6] DAIAN, P., PASS, R., AND SHI, E. Snow white: Robustly reconfigurable consensus and applications to provably secure proof of stake. In *International Conference on Financial Cryptography and Data Security* (2019), Springer, pp. 23–41.

[7] DE VRIES, A. Bitcoin's growing energy problem. *Joule 2*, 5 (2018), 801–805.

[8] DITTMAR, L., AND PRAKTIKNJO, A. Could bitcoin emissions push global warming above 2° c? *Nature Climate Change 9*, 9 (2019), 656–657.

[9] DUTTA, A., BHATTACHARYA, A., KREIDL, O. P., GHOSH, A., AND DASGUPTA, P. Multi-robot informative path planning in unknown environments through continuous region partitioning. *International Journal of Advanced Robotic Systems 17*, 6 (2020), 1729881420970461.

[10] DUTTA, A., GHOSH, A., AND KREIDL, O. P. Multi-robot informative path planning with continuous connectivity constraints. In *2019 International Conference on Robotics and Automation (ICRA)* (2019), IEEE, pp. 3245–3251.

[11] DUTTA, A., PATRICK KREIDL, O., AND O'KANE, J. M. Opportunistic multi-robot environmental sampling via decentralized markov decision processes. In *International Symposium Distributed Autonomous Robotic Systems* (2021), Springer, pp. 163–175.

[12] DUTTA, A., ROY, S., KREIDL, O. P., AND BÖLÖNI, L. Multi-robot information gathering for precision agriculture: Current state, scope, and challenges. *IEEE Access 9* (2021), 161416–161430.

[13] DWORK, C., AND NAOR, M. Pricing via processing or combatting junk mail. In *Annual international cryptology conference* (1992), Springer, pp. 139–147.

[14] GUPTA, M., ABDELSALAM, M., KHORSANDROO, S., AND MITTAL, S. Security and privacy in smart farming: Challenges and opportunities. *IEEE Access 8* (2020), 34564–34584.

[15] HOLLINGER, G. A., AND SINGH, S. Multirobot coordination with periodic connectivity: Theory and experiments. *IEEE Transactions on Robotics 28*, 4 (2012), 967–973.

[16] HOLLINGER, G. A., AND SUKHATME, G. S. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research 33*, 9 (2014), 1271–1287.

[17] JAKOBSSON, M., AND JUELS, A. Proofs of work and bread pudding protocols. In *Secure information networks*. Springer, 1999, pp. 258–272.

[18] KAUFMANN, L. Clustering by means of medoids. In *Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987* (1987), pp. 405–416.

[19] KEMNA, S., ROGERS, J. G., NIETO-GRANDA, C., YOUNG, S., AND SUKHATME, G. S. Multi-robot coordination through dynamic voronoi partitioning for informative adaptive sampling in communication-constrained environments. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017), IEEE, pp. 2124–2130.

[20] KRAUSE, A., SINGH, A., AND GUESTRIN, C. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research 9*, Feb (2008), 235–284.

[21] KRISHNA, C. L., AND MURPHY, R. R. A review on cybersecurity vulnerabilities for unmanned aerial vehicles. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)* (2017), IEEE, pp. 194–199.

[22] LAURI, M., HEINÄNEN, E., AND FRINTROP, S. Multi-robot active information gathering with periodic communication. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017), IEEE, pp. 851–856.

[23] LI, W., ANDREINA, S., BOHLI, J.-M., AND KARAME, G. Securing proof-of-stake blockchain protocols. In *Data Privacy Management, Cryptocurrencies and Blockchain Technology*. Springer, 2017, pp. 297–315.

[24] LUO, W., AND SYCARA, K. Adaptive sampling and online learning in multi-robot sensor coverage with mixture of gaussian processes. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (2018), IEEE, pp. 6359–6364.

[25] MA, K.-C., LIU, L., AND SUKHATME, G. S. Informative planning and online learning with sparse gaussian processes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)* (2017), IEEE, pp. 4292–4298.

[26] MA, K.-C., MA, Z., LIU, L., AND SUKHATME, G. S. Multi-robot informative and adaptive planning for persistent environmental monitoring. In *Distributed Autonomous Robotic Systems*. Springer, 2018, pp. 285–298.

[27] MOGILI, U. R., AND DEEPAK, B. Review on application of drone systems in precision agriculture. *Procedia computer science 133* (2018), 502–509.

[28] NAKAMOTO, S. Bitcoin: A peer-to-peer electronic cash system. Tech. rep., Manubot, 2019.

[29] RASMUSSEN, C. E. Gaussian processes in machine learning. In *Summer School on Machine Learning* (2003), Springer, pp. 63–71.

[30] SAID, T., WOLBERT, J., KHODADADEH, S., DUTTA, A., KREIDL, O. P., BÖLÖNI, L., AND ROY, S. Multi-robot information sampling using deep mean field reinforcement learning. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)* (2021), IEEE, pp. 1215–1220.

[31] SALIMITARI, M., CHATTERJEE, M., AND FALLAH, Y. P. A survey on consensus methods in blockchain for resource-constrained iot networks. *Internet of Things* (2020), 100212.

[32] SAMMAN, T., DUTTA, A., KREIDL, O. P., ROY, S., AND BÖLÖNI, L. Secure multi-robot information sampling with periodic and opportunistic connectivity. In *2022 International Conference on Robotics and Automation (ICRA)* (2022), IEEE, pp. 4951–4957.

[33] SAMMAN, T., SPEARMAN, J., DUTTA, A., KREIDL, O. P., ROY, S., AND BÖLÖNI, L. Secure multi-robot adaptive information sampling. In *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)* (2021), IEEE, pp. 125–131.

[34] SINGH, A., KRAUSE, A., GUESTRIN, C., AND KAISER, W. J. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research 34* (2009), 707–755.

[35] SINGH, A., KRAUSE, A., AND KAISER, W. J. Nonmyopic adaptive informative path planning for multiple robots. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009* (2009), C. Boutilier, Ed., pp. 1843–1850.

[36] STROBEL, V., CASTELLÓ FERRER, E., AND DORIGO, M. Blockchain technology secures robot swarms: a comparison of consensus protocols and their resilience to byzantine robots. *Frontiers in Robotics and AI 7* (2020), 54.

[37] VISERAS, A., WIEDEMANN, T., MANSS, C., MAGEL, L., MUELLER, J., SHUTIN, D., AND MERINO, L. Decentralized multi-agent exploration with online-learning of gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)* (2016), IEEE, pp. 4222–4229.

[38] VISERAS, A., XU, Z., AND MERINO, L. Distributed multi-robot cooperation for information gathering under communication constraints. In *2018 IEEE International Conference on Robotics and Automation (ICRA)* (2018), IEEE, pp. 1267–1272.

[39] VORONOI, G. New applications of continuous parameters ‘a to the theory of quadratic forms. second memory. research on primitive parallels. *Journal f ”u r die Reine und angewandte Mathematik (Crelles Journal) 1908*, 134 (1908), 198–287.

[40] WEI, Y., AND ZHENG, R. Informative path planning for mobile sensing with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications* (2020), IEEE, pp. 864–873.

VITA

Tamim Khatib has a Bachelor of Science in Computer Science and Information Sciences on a Computer Science track from the University of North Florida (UNF). Tamim expects to receive a Master of Science in Computer Science from UNF by December 2022. Dr. Ayan Dutta is serving as Tamim's thesis advisor. Tamim is currently employed as an IT technician for First Coast Cardiovascular Institute and has been there since June 2020, and he is employed as a researcher at UNF since January 2021. Before that, he was a Teaching Assistant (SI Instructor) for UNF's Computation Structures course from September 2018 to December 2019.

Tamim's research work has largely focused on secure multi-robot information sampling and multi-robot connectivity modeling. His first paper, *Secure Multi-Robot Adaptive Information Sampling* was published at the 2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR). His second paper, *Secure Multi-Robot Information Sampling with Periodic and Opportunistic Connectivity* was presented at the IEEE International Conference on Robotics and Automation (ICRA) in May 2022.