Secure Multi-Robot Information Sampling with Periodic and Opportunistic Connectivity

Tamim Samman, Ayan Dutta, O. Patrick Kreidl, Swapnoneel Roy, Ladislau Bölöni

***Abstract*— Multi-robot teams are becoming an increasingly popular approach for information gathering in large geographic areas, with applications in precision agriculture, surveying the aftermath of natural disasters or tracking pollution. These robot teams are often assembled from untrusted devices not owned by the user, making the maintenance of the integrity of the collected samples an important challenge. Furthermore, such robots often operate under conditions of opportunistic, or periodic connectivity and are limited in their energy budget and computational power. In this paper, 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 tradeoffs between data integrity, model accuracy, and time consumption.**

I. Introduction

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 [9], [12]. 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. Similar to the way blockchain technology allows the creation of a trusted ledger over the internet, several recent projects explore the application of blockchain-based technology to ensure the trustworthiness of agricultural information.

Unfortunately, algorithms used for cryptocurrencies do not directly apply to teams of agricultural 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, 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).

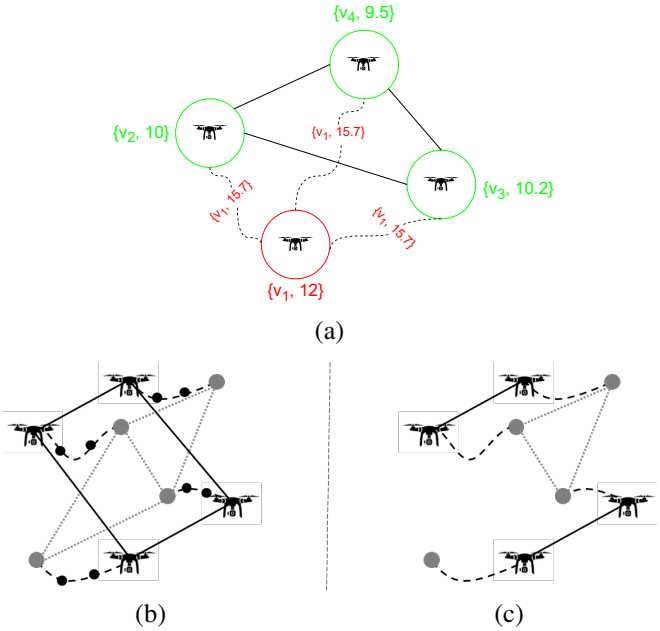

Fig. 1. (a) An instance of a data integrity attack studied in [26] using blockchain-based security techniques under continuous connectivity (CC) assumptions; an assumed malicious robot (red circled) may send tampered data to its neighbor robots (green circled) in order to degrade future estimates of the underlying information field. We consider the problem when communication resources are more constrained, specifically under scenarios of (b) periodic connectivity (PC) in which robots share measurements only in a subset of the rounds (e.g., every three rounds) and (c) opportunistic connectivity (OC) in which robots share measurements possibly only among a subset of the nodes (e.g., those within range).

This paper describes blockchain-based techniques that ensure the integrity of the data in the context of the multi-robot information sampling problem: given n mobile robots and a budget B, plan B-length paths for the robots such that the collected information is maximized [9], [10], [21], [27]. Each robot is equipped with an information collection sensor (e.g., camera) 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 that integrates all (trusted) observations made by the robots. To plan such optimal paths is proven to be NP-Hard and, therefore, greedy heuristics for navigation are popularly employed [9], [12], [16], [30]. The first work to integrate blockchain-based security and multi-robot information collection, observing the degree to which robots must tradeoff between integrity

T. Samman, A. Dutta, O.P. Kreidl, and S. Roy are with the University of North Florida, USA. Emails: {n01379084, a.dutta, patrick.kreidl, s.roy}@unf.edu
L. Bölöni is with the University of Central Florida, USA. Email: {ladislau.boloni}@ucf.edu
This work is supported in part by NSF CPS Grant #1932300

guarantees and energy consumption [26], was developed entirely under continuous connectivity assumptions against a single attacker. This paper considers scenarios in which the robots only have access to periodic or opportunistic connectivity and performances of the proposed algorithms are investigated on a variety of scenarios with single or multiple attackers (Fig. 1). The main contributions are:

1) To the best of our knowledge, we are the first to consider the problem of maintaining data integrity in multi-robot informative path planning with periodic or opportunistic connectivity.
2) We propose and validate practical data integrity algorithms based on blockchain that are specifically designed to be deployable on robots.
3) We study the algorithms along the novel perspective of the tradeoffs between data integrity, energy consumption and model estimate error.

II. Background

Information gathering using a single or multiple robots has received considerable attention in the recent literature [4], [8], [9], [10], [19], [18], [31]. 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 get collected by human operators [13], [21], [27], [32]. On the other hand, in a life-long learning and sampling scenario such as studied in this paper, the robot(s) can be given a budget for one particular day's mission and the objective is to collect the maximum information possible within the daily budget [8], [19], [24]. These information gathering techniques often use 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 the maximum [20], [27], [28]. 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 [15], [19], [10], [26]. In [10], the authors have proposed a decentralized MDP-based online coordination mechanism that lets the robot collect maximal information even under control uncertainty. Only recently, researchers have started looking into reinforcement learning-based information sampling technique while using Gaussian Process as the information modeling tool [24], [32].

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. However, keeping continuous connectivity (CC) in a dynamic, online informative path planning scenario has been proved to be compute-intensive [9]. On the other hand, if the robots want 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], [12], [18]. The final connectivity technique is opportunistic (OC), where the robots coordinate with others only when two or more are in vicinity. As this does not pose any connectivity policy restrictions, this strategy is often adopted in general multi-robot exploration studies [2], [8], [10]. A survey of various connectivity models for multi-robot exploration and coordination can be found in [1].

None of the above-mentioned works consider adversary influence, such as tampering with measured data, on multi-robot information collection. 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 so-called Proof-of-Work (PoW) consensus was examined in [29] and [26], each assuming the robots enjoyed continuous connectivity (CC). PoW consensus is a popular and effective choice in the cryptocurrency industry [22], but it is known to be resource intensive [6], [7] and, in turn, raises new challenges for resource-limited multi-robot information collection.

III. Problem Setup

We have a set of n homogeneous robots $R = r_1, r_2, \cdots, r_n$ that explore a shared environment. The environment is discretized into a planar 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 sub-region for exploration, $\mathcal{V}_i$, and $\mathcal{V}_i \cap \mathcal{V}_j = \emptyset$. We have pre-calculated $\mathcal{V}_i$ using K-medoids clustering [14]. r_i is equipped with an on-board sensor using which it can sense and collect information (e.g., radiation detector). 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.

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 [23]. The volumetric measure of this uncertainty is calculated by an information theoretic metric, (differential) entropy, which is formally defined as $H(\mathbf{X}) = \frac{1}{2}\log|\Sigma| + \frac{|V|}{2}\log(2\pi e)$. 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:

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

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. During periodic or opportunistic connectivity, the

posterior statistics will sometimes evolve on a batch of measurements, which is easily accommodated by appropriate augmentation of the output matrix $\mathbf{C}(\cdot)$—the reader is referred to [10] for more details. It is a standard assumption in kernel-based parametrizations of GPs that the the correlation between two nodes are inversely proportional to the distances between them [10], [16], [23]. 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 given by

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

The next section utilizes 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. 2. In a greedy fashion, r_i then chooses the next adjacent node $v_i^* \in \mathcal{V}_i$ that provides the maximum information.

$$v_i^* = \arg\max_{v \in adj(v_i^0)} H(v|\mathcal{D} \cup \mathcal{Z}(v_i^0)),\ \text{s.t. } v \in \mathcal{V}_i \tag{3}$$

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], [16], [27].

IV. Algorithms

It is not this paper's objective 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 [30], [1], [9], [26]. This paper is interested in studying the resilience against data integrity attacks within the constrained communication setting, specifically under periodic (PC) and opportunistic connectivity (OC). In PC, the robots will form a connected network after every $\mathcal{F}$ cycles named coordination frequency [18], [12]. The readers are referred to [3], [12] to see how such re-connections 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 [8], [10].

A. Proof-of-Work (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. 1 (e.g., see Algorithm 1). One or more malicious entities can attack this data sharing system via data tampering attempts and denial-of-service (DoS) [11], [17]. To prevent other robots to incorporate such fake data for their future decision-making, we have used a Blockchain-based security protocol. Blockchain is a tamper-resistant digital ledger that the robots maintain in a distributed fashion [25]. 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. Similar to [26], 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 blockchain because of its chain data structure – if an attacker is 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}.

Algorithm 1: Secure Information Sampling

```
/* 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 Z(v_i^*);
    Create a block b_idx that includes v_i^* and Z(v_i^*);
    Add b_idx to C_i and broadcast it 1) every cycle with
      OC, or 2) periodically every F cycle with PC;
    C̃ ← receive similar blockchains from 1) ∀r_j ∈ R \ r_i
      with PC, or 2) ∀r_j ∈ R̄ with OC;
    Secure. Decide to add the measurements from C̃ to C_i
      or not using PoW [26, Algorithm 2];
    Estimate. update GP_i with the new data in C_i (Eq. 1)
      and update the entropies (Eq. 2);
    Select v_i^* based on the updated entropies (Eq. 3);
```

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 are there 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 [26, Algo. 2]. Note that the verification of

the hash is straightforward. A robot looks at the nonce in a particular block, finds it corresponding hash, and checks whether the hash has the desired difficulty level. If not, the block is rejected; otherwise, it is validated. As can be understood, increasing the difficulty reduces the probability of it being compromised while the time and energy required by the robots increase significantly.

B. 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 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. ■

V. Experiments

We have implemented our 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 $\mathcal{V}_i$. Our adversary model is the same as our earlier work in [26], where a data-tampering adversary (e.g., one of the robots in the system) falsifies its measurement only every four rounds. The random false measurement is chosen in the range of $[-10, +10]$. For more details, the readers are referred to [26]. Unlike [26], our experiments include the case of multiple adversarial robots.

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 $\beta^2 \exp(-||v_s - v_t||/\ell)$, 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 $\epsilon \in \mathcal{N}(0, 0.25)$. We compare our proposed PoW-based algorithms with CC, PC, and OC assumptions against two benchmarks: 1) *No Attack*. in this scenario, there is no malicious robot in the system, and therefore, there is no chance of data tampering; and 2) *Insecure*. in this scenario, data integrity attacks are exactly similar to the attacks on our algorithms, however, there is no security protocol in place to protect against such attacks.

1) Single Attacker: 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. 2 and 3.(a). The results for CC [26] are also presented to benchmark the PC and OC results against it. When we compare MSE results with PC against the insecure version, it almost always performs statistically significantly better. Similar to CC, the the blockchain-based proposed technique will fail to safeguard against the data tampering attempts if the selected difficulty is low, e.g., 1. 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 $\frac{1}{16}$, which is fairly low, and therefore, the malicious robot can tamper the global data sharing once in a while. When we compare the PC results with varying $\mathcal{F}$, the robots performed better, i.e., final MSE was lower, 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. 3.(a) for reference). In general, the closer the connectivity model is to CC, the better performance in terms of MSE can be observed because of the reason stated above. Note that, this results in a higher computation time, which we will discuss later in the section.

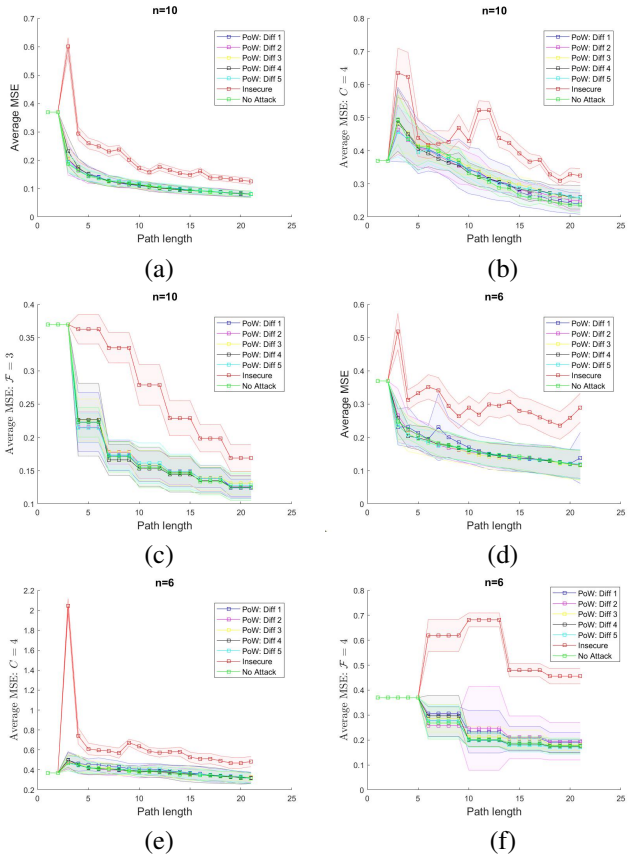

Fig. 2. Single Attacker: MSE comparison (lower the better) among various connectivity models used: (a,d) CC with $n = 10$ and 6; (b,e) OC with $n = 10$ and 6; and (d,f) PC with $n = 10$ and 6.

Similar to CC and PC, the OC model almost always performs statistically significantly better than the insecure version except a few cases with difficulty 1 due to reason explained 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 are significant. For example, with difficulty 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

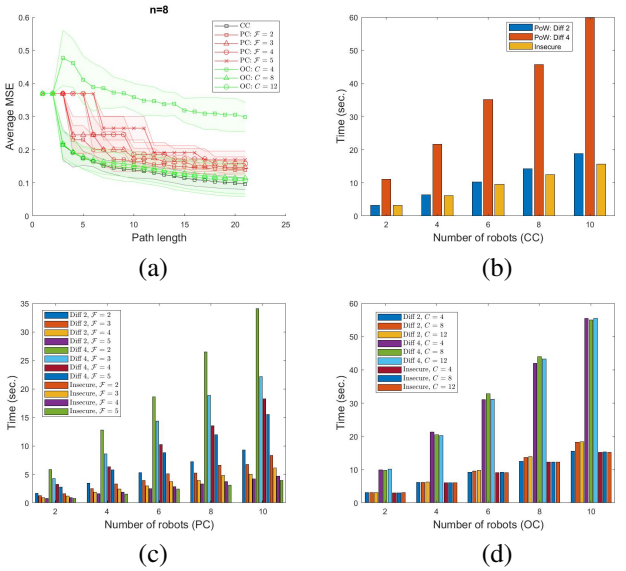

Fig. 3. Single Attacker: (a) Comparison of MSE values among all the connectivity models with $n = 8$; Run time comparison (lower the better) between our proposed secure techniques and the implemented benchmark algorithms: (b) CC; (c) PC; and (d) OC.

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 range 8 seem more similar to range 4 than $C = 12$. This is because that 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). The robots that communicated more often usually had less data that needed to redo PoW, and consequently, required less time.

When compared all three connectivity models together (Fig. 3.(a)), CC always performed the best. Both PC and OC performed significantly worse when facing more constrained conditions, i.e., higher coordination frequencies for PC and lower C values for OC. In particular, OC performed among the three connectivity models when there were few robots since that meant they would rarely communicate. PC always performed worse than CC since the robots with CC always communicate and coordinate more often, i.e., after every data collection, and therefore, the robots could adapt their joint paths on a finer scale. For OC, however, this is not the case. When the C and n are high, OC becomes almost identical to CC as all the robots can share their collected data after every round and fine tune their paths.

In regards to the run time of the algorithm, PC always outperformed CC (Fig. 3.(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 sec. respectively while the run time for CC is 59.76 sec. On the other hand, OC always performed better than CC, but worse than PC.

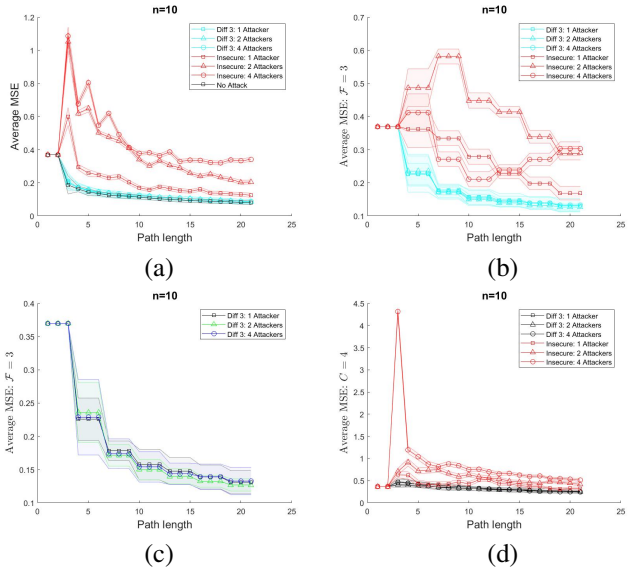

Fig. 4. Multiple Attackers: MSE comparison (lower the better) among various connectivity models used: (a) CC; (b) PC; (c) PC zoomed in on our algorithm's results; and (d) OC.

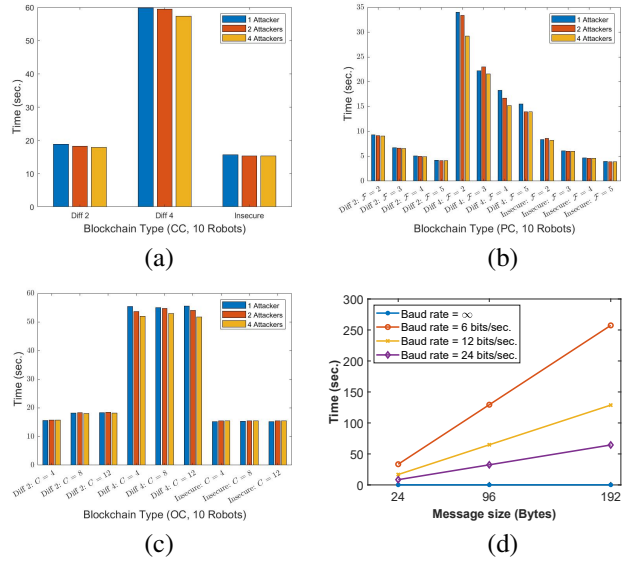

Fig. 5. Run time comparison (lower the better) among various connectivity models used with Multiple Attackers: (a) CC; (b) PC; (c) OC; and (d) Effect of various Baud Rates and message sizes on run time ($n = 2$).

Additionally, while OC usually did better when range was greater, in some case, this is not followed. This is because that there were certain occasions when the time saved from coordinating less often was counterbalanced by the time spent in re-doing PoW for the orphaned blocks.

2) Multiple Attackers: For CC, PC, and OC, the difference in finals MSEs with 1, 2, and 4 malicious robots was small (Figs. 4.(b-d)). As the objective of PoW is to ensure that attacks do not affect accuracy, therefore, 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 $\mathcal{F}$ set to 3 in PC, the final MSE is 0.131, whereas with four attacking robots this value is 0.133. This 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. More malicious robots in the system have led to a decrease in run time. While PoW takes up the majority of the computation time [26], 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.

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 have 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 containing a block is smaller when compared to PC, where the robots share past $\mathcal{F}$ collected data in a single message. The result is presented in Fig. 5. When the baud rate is set to infinity, a standard assumption in multi-robot coordination studies [9], [8], [31], [15], [10], [19], the communication time is almost negligible, the maximum being 0.13 sec. On the other hand, when it is restricted to be only 6 bits/sec., to send a 192 byte-size message (e.g., putting past eight observations in message), it takes 257.40 sec. whereas for a 24 byte message, the communication time is 33.40 sec. This result is significant in terms of CC, PC, and OC comparisons. Although PC needed a fraction of 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, the robots might need to exchange large chunks of data if and when they come within each other's communication ranges.

VI. Conclusion and Future Work

This paper proposes a method aiming to improve data integrity for a multi-robot team performing informative path planning under periodic and opportunistic connectivity. The approach builds on blockchain technology adapted to the connectivity and energy limitations. We performed an extensive set of experiments assuming threat models with 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. Future work will 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.

REFERENCES

[1] F. Amigoni, J. Banfi, and N. Basilico. Multirobot exploration of communication-restricted environments: a survey. *IEEE Intelligent Systems*, 32(6):48–57, 2017.

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

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

[4] J. Banfi, A. Q. Li, I. Rekleitis, F. Amigoni, and N. Basilico. Strategies for coordinated multirobot exploration with recurrent connectivity constraints. *Autonomous Robots*, 42(4):875–894, 2018.

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

[6] A. De Vries. Bitcoin's growing energy problem. *Joule*, 2(5):801–805, 2018.

[7] L. Dittmar and A. Praktiknjo. Could bitcoin emissions push global warming above 2° c? *Nature Climate Change*, 9(9):656–657, 2019.

[8] A. Dutta, A. Bhattacharya, O. P. Kreidl, A. Ghosh, and P. Dasgupta. Multi-robot informative path planning in unknown environments through continuous region partitioning. *International Journal of Advanced Robotic Systems*, 17(6):1729881420970461, 2020.

[9] A. Dutta, A. Ghosh, and O. P. Kreidl. Multi-robot informative path planning with continuous connectivity constraints. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 3245–3251. IEEE, 2019.

[10] A. Dutta, O. Patrick Kreidl, and J. M. O'Kane. Opportunistic multi-robot environmental sampling via decentralized markov decision processes. In *International Symposium Distributed Autonomous Robotic Systems*, pages 163–175. Springer, 2021.

[11] M. Gupta, M. Abdelsalam, S. Khorsandroo, and S. Mittal. Security and privacy in smart farming: Challenges and opportunities. *IEEE Access*, 8:34564–34584, 2020.

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

[13] G. A. Hollinger and G. S. Sukhatme. Sampling-based robotic information gathering algorithms. *The International Journal of Robotics Research*, 33(9):1271–1287, 2014.

[14] L. Kaufmann. Clustering by means of medoids. In *Proc. Statistical Data Analysis Based on the L1 Norm Conference, Neuchatel, 1987*, pages 405–416, 1987.

[15] S. Kemna, J. G. Rogers, C. Nieto-Granda, S. Young, and G. S. Sukhatme. 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)*, pages 2124–2130. IEEE, 2017.

[16] A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research*, 9(Feb):235–284, 2008.

[21] K.-C. Ma, Z. Ma, L. Liu, and G. S. Sukhatme. Multi-robot informative and adaptive planning for persistent environmental monitoring. In *Distributed Autonomous Robotic Systems*, pages 285–298. Springer, 2018.

[17] C. L. Krishna and R. R. Murphy. A review on cybersecurity vulnerabilities for unmanned aerial vehicles. In *2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR)*, pages 194–199. IEEE, 2017.

[18] M. Lauri, E. Heinänen, and S. Frintrop. Multi-robot active information gathering with periodic communication. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 851–856. IEEE, 2017.

[19] W. Luo and K. Sycara. 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)*, pages 6359–6364. IEEE, 2018.

[20] K.-C. Ma, L. Liu, and G. S. Sukhatme. Informative planning and online learning with sparse gaussian processes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4292–4298. IEEE, 2017.

[22] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. Technical report, Manubot, 2019.

[23] C. E. Rasmussen. Gaussian processes in machine learning. In *Summer School on Machine Learning*, pages 63–71. Springer, 2003.

[24] T. Said, J. Wolbert, S. Khodadadeh, A. Dutta, O. P. Kreidl, L. Bölöni, and S. Roy. Multi-robot information sampling using deep mean field reinforcement learning. In *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1215–1220. IEEE, 2021.

[25] M. Salimitari, M. Chatterjee, and Y. P. Fallah. A survey on consensus methods in blockchain for resource-constrained iot networks. *Internet of Things*, page 100212, 2020.

[26] T. Samman, J. Spearman, A. Dutta, O. P. Kreidl, S. Roy, and L. Bölöni. Secure multi-robot adaptive information sampling. In *2021 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 125–131, 2021.

[27] A. Singh, A. Krause, C. Guestrin, and W. J. Kaiser. Efficient informative sensing using multiple robots. *Journal of Artificial Intelligence Research*, 34:707–755, 2009.

[28] A. Singh, A. Krause, and W. J. Kaiser. Nonmyopic adaptive informative path planning for multiple robots. In C. Boutilier, editor, *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence, Pasadena, California, USA, July 11-17, 2009*, pages 1843–1850, 2009.

[29] V. Strobel, E. Castelló Ferrer, and M. Dorigo. Blockchain technology secures robot swarms: a comparison of consensus protocols and their resilience to byzantine robots. *Frontiers in Robotics and AI*, 7:54, 2020.

[30] A. Viseras, T. Wiedemann, C. Manss, L. Magel, J. Mueller, D. Shutin, and L. Merino. Decentralized multi-agent exploration with online-learning of gaussian processes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4222–4229. IEEE, 2016.

[31] A. Viseras, Z. Xu, and L. Merino. Distributed multi-robot cooperation for information gathering under communication constraints. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1267–1272. IEEE, 2018.

[32] Y. Wei and R. Zheng. Informative path planning for mobile sensing with reinforcement learning. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 864–873. IEEE, 2020.