Byzantine Resilience at Swarm Scale: A Decentralized Blocklist Protocol from Inter-robot Accusations

Kacper Wardega
Boston University
Boston, USA
ktw@bu.edu

Max von Hippel
Northeastern University
Boston, USA
vonhippel.m@northeastern.edu

Roberto Tron
Boston University
Boston, USA
tron@bu.edu

Cristina Nita-Rotaru
Northeastern University
Boston, USA
c.nitarotaru@northeastern.edu

Wenchao Li
Boston University
Boston, USA
wenchao@bu.edu

ABSTRACT

The Weighted-Mean Subsequence Reduced (W-MSR) algorithm, the state-of-the-art method for Byzantine-resilient design of decentralized multi-robot systems, is based on discarding outliers received over Linear Consensus Protocol (LCP). Although W-MSR provides theoretical guarantees relating network connectivity to the convergence of the underlying consensus, W-MSR comes with several limitations: the number of Byzantine robots, F, to tolerate should be known a priori, each robot needs to maintain $2F + 1$ neighbors, $F + 1$ robots must independently make local measurements of the consensus property in order for the swarm's decision to change, and W-MSR is specific to LCP and does not generalize to applications not implemented over LCP. In this work, we propose a Decentralized Blocklist Protocol (DBP) based on inter-robot accusations. Accusations are made on the basis of locally-made observations of misbehavior, and once shared by cooperative robots across the network are used as input to a graph matching algorithm that computes a blocklist. DBP generalizes to applications not implemented via LCP, is adaptive to the number of Byzantine robots, and allows for fast information propagation through the multi-robot system while simultaneously reducing the required network connectivity relative to W-MSR. On LCP-type applications, DBP reduces the worst-case connectivity requirement of W-MSR from $(2F + 1)$-connected to $(F + 1)$-connected and the minimum number of cooperative observers required to propagate new information from $F + 1$ to just 1 observer. We demonstrate that our approach to Byzantine resilience scales to hundreds of robots on target tracking, time synchronization, and localization case studies.

KEYWORDS

Multi-Robot Systems; Multi-Robot System Security; Byzantine-Resilient Swarms

ACM Reference Format:
Kacper Wardega, Max von Hippel, Roberto Tron, Cristina Nita-Rotaru, and Wenchao Li. 2023. Byzantine Resilience at Swarm Scale: A Decentralized Blocklist Protocol from Inter-robot Accusations. In *Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), London, United Kingdom, May 29 – June 2, 2023*, IFAAMAS, 9 pages.

Proc. of the 22nd International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2023), A. Ricci, W. Yeoh, N. Agmon, B. An (eds.), May 29 – June 2, 2023, London, United Kingdom. © 2023 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

1 INTRODUCTION

Multi-robot systems (MRS) presently employed in industry use structured deployment environments and highly centralized designs [27]. Central coordination benefits all key MRS components – task allocation, execution, fault detection and recovery,while structured environments allow for strict physical security measures. In contrast, emergent MRS applications in unstructured environments (such as patrol, search and rescue, coverage, shape formation, and collective transport) are typically not amenable to centralized approaches due to communication constraints [11]. Decentralized methods to mitigate the negative impact of faulty and/or malicious robots in unstructured environments have therefore attracted much research attention, especially since a wide range of attacks have been shown to disrupt MRS function and safety, e.g. sensor perturbation and denial-of-service (DoS) [7, 16, 31], actuator jamming [14], networking DoS [29], or Sybil/fraudulent identity attacks [12, 17]. Given the multitude of possible attacks, it is important to understand the resilience of the MRS from *Byzantine attackers* – that is if an unknown subset of the robots is allowed to have arbitrarily different behaviors relative to the cooperative robots in terms of physical actions and communication.

Byzantine-unaware MRS implementations are often highly vulnerable, and break completely, when even one robot has been comprised. In our case studies for example, Byzantine robots may cause robots within a swarm to follow a false target, or have arbitrarily large errors in time synchronization or localization. The main approach proposed for Byzantine-resilient MRS is the *Weighted-Mean Subsequence Reduced* (W-MSR) algorithm [15, 18, 24]. W-MSR is easy to implement and has well-understood theoretical guarantees. However, W-MSR can only be used for MRS applications that are implemented via Linear Consensus Protocol (LCP), performance does not scale with the number of robots in the system, and the number of Byzantine robots to tolerate, F, is a parameter that must be known a priori. Suppose that LCP is the means by which the robots reach a collective decision about a physical property of the environment. The choice of F in W-MSR dictates how many outliers robots should discard in each update of linear consensus; each robot needs at minimum $2F + 1$ neighbors in order to update their local consensus variable and at minimum $F + 1$ cooperative robots must independently make direct measurement of the underlying physical quantity. If F is chosen smaller than the number of Byzantine robots, then the mitigation provided by W-MSR is forfeit. For large F, the

network connectivity requirement and the logistics of maintaining $F + 1$ cooperative observers renders W-MSR impractical.

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

2 BACKGROUND & RELATED WORK

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

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

and $\mathcal{N}_G(i)$ are the neighbors of i in the connectivity graph G. The authors of [15] first introduced W-MSR for Byzantine resilience which discards the F highest and F lowest values received at each timestep of LCP, and show that convergence despite up to F Byzantine robots is equivalent to a graph robustness property. Specifically, if the connectivity graph of the robots is at least $(2F + 1)$-vertex-connected, then the consensus will converge to a value within the convex hull of the cooperative robots' initial values. W-MSR has been applied to a variety of applications, such as flocking [24] and state estimation [18]. Extensions for the W-MSR algorithm to time-varying networks where the union of the connectivity graphs within a bounded window is robust are proposed in [23] and to event-driven control in [2]. Methods to form robust graph topologies, as required by the W-MSR algorithm, are proposed in [13].

Blockchain. Distributed ledger technologies, e.g. blockchains, have also attracted much research attention for its potential to provide resilience guarantees. For similar settings as considered in this paper, [26] proposes to use an Ethereum blockchain for a MRS collective decision-making case study. The authors of [19] investigate the approach of consensus over blockchain. We refer the reader to a recent survey [1] of work on blockchain for robotics applications, including for MRS and swarm.

Inter-robot observations. The use of inter-robot observations to detect misbehavior and establish trust is a common theme in multi-agent systems generally. As opposed to our work, where accusations are used to compute a blocklist, the authors of [4] propose that cooperative robots should take physical action to isolate misbehaving robots on observing incorrect behavior. In a cooperative patrolling case study for example, cooperative robots surround and impede the movement of robots that are observed not following the correct trajectory. More commonly, inter-robot observations are used as input to a reputation mechanism, whereby robots maintain real-valued reputation scores of their peers. For example, [10] presents a connected vehicle case study where robots use partial information to determine if their local neighbors are non-cooperative, and a later work [3] proposes an adaptive threshold-based actuator fault detection strategy for MRS. The use of reputation scores as input to a cooperative coverage problem is explored in [20], and [6] introduces a general trust framework for multi-robot systems with case studies on intelligent intersection management. Reputation mechanisms can also be merged with consensus, i.e. [9] proposes that robots perform consensus on the reputation values of the robots. Ultimately, reputation mechanisms inherit the drawbacks of W-MSR, in that $F + 1$ cooperative robots would be required to assign a low reputation to each misbehaving robot before their influence is removed from the system, and furthermore the number of consensus variables (the reputation scores) scales linearly with the size of the swarm.

MRS security. Beyond the mentioned directly related works, our work is motivated by the broader push for secure MRS. We refer the reader to [5] for a comprehensive survey of open security problems in MRS, with problem-specific recommendations for mitigation, and to [30] for a treatment of recent approaches to MRS in uncertain or adversarial operating environments. Decentralized MRS has a wide attack surface; [14] tests CPS-inspired anomaly detection for a variety of compromise scenarios such as wheel jamming, LiDAR denial-of-service, wheel encoder logic fault, etc., and [7] analyzes bounded sensor attacks. A novelty of our work is the proof-of-concept Byzantine-resilience for cooperative localization. Byzantine localization is a challenging problem, [28] derives conditions under which Byzantine localization is solvable in the centralized setting; similarly as with W-MSR, Byzantine localization requires the graph structure of inter-robot measurements to satisfy a robustness property.

Sybil attacks. Certain threat models have received special treatment in the literature. Efficient methods for scheduling MRS under denial of service attacks are proposed in [31], which [16] extends to the decentralized setting. In our work, we assume that the MRS is protected from Sybil attacks since a central trust authority issues identities for all of the robots. We believe that Sybil-proofness of the system via central authority is a reasonable assumption for a *closed* MRS, where the set of robots does not change with time. Decentralized identity management and Sybil-proofness for MRS is however an active area of research orthogonal to our own. Using

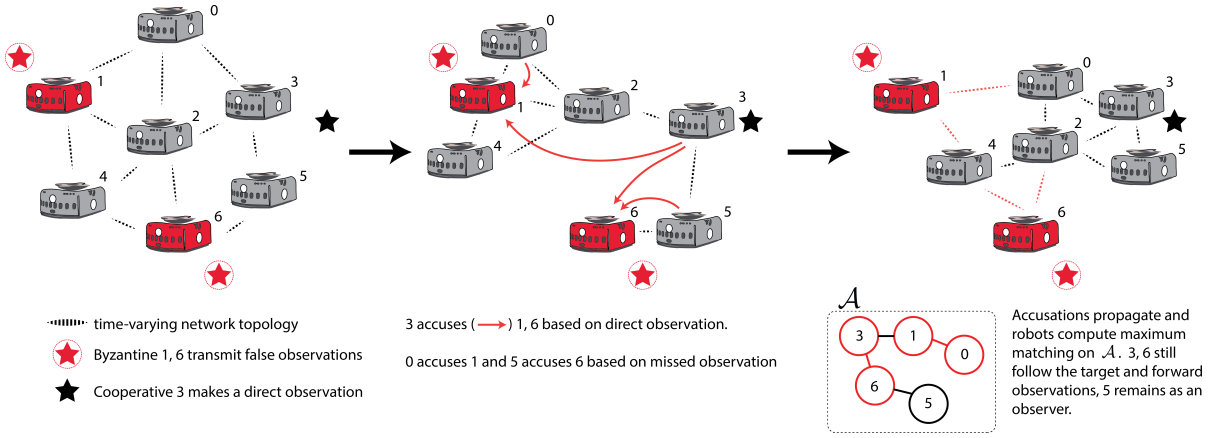

Figure 1: DBP is used to provide Byzantine resilience for a simple seven robot (two Byzantine) scenario of the target tracking case study explored in Section 5.1. Byzantine robots 1 and 6 transmit false target observations (red stars) while cooperative robot 3 makes a direct observation of the target (black star). Based on the closest observer, the cooperative robots move towards the supposed target location. Robots 0 and 5 make it to the false locations reported by 1 and 6 respectively and accuse the observers on the basis of missed observations that should have been made. Meanwhile, robot 3 accuses both 1 and 6 since their observations contradict 3's direct observation. The accusations flood through the cooperative robots, until eventually each cooperative robot has accusation graph denoted $\mathcal{A}$. Edmond's algorithm is used by each robot to independently compute the maximum matching $\{(3,6),(0,1)\}$ (red edges), thus observations from robots 0, 1, 3, and 6 are blocked. Observations from robots 2, 4, and 5 are still trusted by the other cooperative robots. Note that 0 and 3 still continue to cooperate by moving towards the target and forwarding observations from non-blocked observers.

inter-robot radio signals to detect Sybil identities is first proposed in [12]; the same technique is later applied to a cooperative flocking scenario under Sybil threat model in [17]. Other physics-inspired, application-specific approaches have been proposed in the literature, for example defending against Sybil attacks on a crowdsourced traffic light [25]. Prior work has also proposed to incorporate Sybil attack prevention as a component of the W-MSR algorithm [22].

3 THREAT MODEL

We consider a swarm robotics system with robots connected by time-varying network topology $G[t] = (V, E[t])$. Each robot has an identity that has been issued by a trusted central authority at deploy-time, which it uses to both send signed messages to its neighbors and to verify the authenticity of received messages. Assume that some unknown subset of the robots have been compromised by a Byzantine adversary. We refer to the cooperative robots as $C \in V$, the Byzantine ones as $\bar{C} = V \setminus C$, and we assume that the sets V and C are fixed, i.e. the MRS is *closed*. We assume a strong adversary, where the Byzantine robots can coordinate centrally with each other online, have detailed knowledge of the system implementation such as robot capabilities and application details, can send arbitrary messages to the cooperative robots, and have arbitrary physical behaviors. The goal of the Byzantine robots is to disrupt the MRS application; the specific goal and attack strategy will depend on the application. Each of our case studies in Section 5 will specify the attacker's goal and strategy. Since the robots use their trust authority-issued identities to communicate, we assume that Sybil attacks are not possible for the adversary, since the adversary is unable to forge fraudulent identities that will be accepted by C. However, robots whose identities (secret keys) have been compromised can be used by the adversary to send misleading messages. Therefore, any robots in the swarm whose keys have been compromised are considered to be part of $\bar{C}$.

4 DECENTRALIZED BLOCKLIST PROTOCOL

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

DBP relies on flooding as a networking primitive, where cooperative robots always re-broadcast (forward) received messages. Messages in DBP are accusations signed by the robot initiating the flood. Accusations $\texttt{Acc}_i(j)$ are an application-agnostic message and the payload is simply the identity of a robot j that the origin i wishes to accuse. The precise rules used to decide if and when an accusation should be issued are application-specific. Accusations serve to remove the influence of Byzantine nodes on the swarm application. Each robot i locally maintains a set $R_i[t]$ of accusations that it has received. A subset $R_i^*[t] \subseteq R_i[t]$ will be locally computed by i using any deterministic maximum matching algorithm (such as Edmond's [8]) to form the blocklist. For the remainder of this section, we will assume that the robots have a *sound* accusation mechanism:

Definition 1 (Sound accusation). *If a cooperative robot i issues an accusation $\mathtt{Acc}_i(j)$, then $\mathtt{Acc}_i(j)$ is* sound *if and only if $j \in \bar{C}$.*

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

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

Definition 2 (Hall's Marriage condition). *Given $G = (X \cup Y, E)$ s.t. G is X-semi-bipartite, a Y-perfect matching exists if $\forall S \subseteq Y$, $|S| \leq |\mathcal{N}_G(S) \cap X|$. Additionally, any maximum matching will be Y-perfect.*

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

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

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

The s-closure of the flood is then the union

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

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

Theorem 1 (Eventual Blocklist Consensus). *Let $G[t]$ be the time-varying, (F, n)-floodable network topology of the robot swarm V. If $|\bar{C}| \leq F$, there $\exists \tau \in \mathbb{Z}^+, \mathcal{A} \forall i \in C, s \geq \tau$ s.t. $\mathcal{A} = \mathcal{A}_i[s]$.*

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

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

5 CASE STUDIES

We run our experiments on turtlebots simulated in ARGoS [21], a multi-physics robot simulator that can efficiently simulate large-scale swarms of robots. The robots are equipped with a radio to transmit to neighboring robots within 4m and have an omnidirectional camera used for nearby target detection and collision avoidance with an observation distance of ≈ 0.9m. The robot controller runs at 30Hz. Source code to reproduce our experiments can be found at https://github.com/gitsper/decentralized-blocklist-protocol

5.1 Target Tracking

Application overview. In swarm target tracking, the goal of the robots is to locate and cooperatively follow a mobile target that has a maximum speed of d. In our experimental setup, the target is a robot that has a yellow light – robots within a distance r can see the light and make a direct observation of the target. To enable the entire swarm to track the target, even for those robots that do not directly observe the target, robots broadcast target observation messages containing:

(1) the observer's unique ID
(2) the time of the observation
(3) the observed location of the target

In each timestep, robots sort received observation messages by observation time, and choose the most recent one to transmit to its neighbors. Robots keep track of how many times a given observation message has been transmitted, and stop sharing it after fixed,

finite number of times. The purpose of transmitting the same observation message multiple times is to account for the time-varying connectivity with neighboring robots. In addition to the application messages, DBP is used to mitigate the influence of Byzantine robots. Robots delete and do not forward observations messages from blocked observer IDs. Old observation messages are periodically deleted from the local cache.

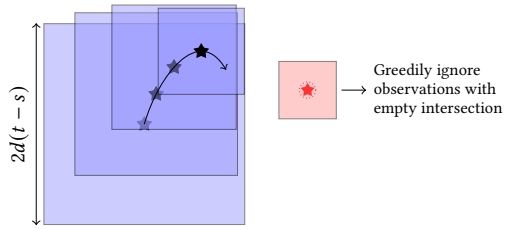

Figure 2: Observation-based target tracking setup for use with DBP. Robots that do not observe the target directly sort received observations by age and compute a bounding box for each observation containing the target based on the elapsed time. Reducing over the bounding boxes with the set intersection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

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

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

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

Accusation rules. On receiving a new observation message, robots issue DBP accusations according to four target tracking-specific accusation rules. Given the received observation by robot j of $\tilde{x}$ made at time s, let $\Delta t = t - s$ the elapsed time, $\Delta p_i = \|p_i[t] - \tilde{x}\|$ the distance from i's location $p_i[t]$ to the observed target, and c a constant denoting an upper bound on the speed with which messages can travel through the network (in our experimental setup, 4m/timestep). The first accusation rule is triggered when $r + c\Delta t < \Delta p_i$, as the observation would need to have traveled faster-than-possible through the network. The second accusation rule is triggered when $\Delta p_i < r - d\Delta t$ and i did not make a direct observation of the target – i missed an observation that it should have made if the received observation was legitimate. The third accusation is rule is triggered when $\Delta p_i > r + d\Delta t$ but a direct observation *was* made by i; in this case the target couldn't possibly have moved fast enough from the received observation location to the place where i observed it presently. Finally, the last accusation rule detects oscillations from a single observer. If i has received an observation from j in the past, it will consider the most recent previous observation from j of $\tilde{x}_{\text{old}}$ at time s_{old}, and will make an accusation of j if $\|\tilde{x} - \tilde{x}_{\text{old}}\| > d(s - s_{\text{old}})$. In this case, j's observations are inherently inconsistent with the maximum rate of change in x.

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

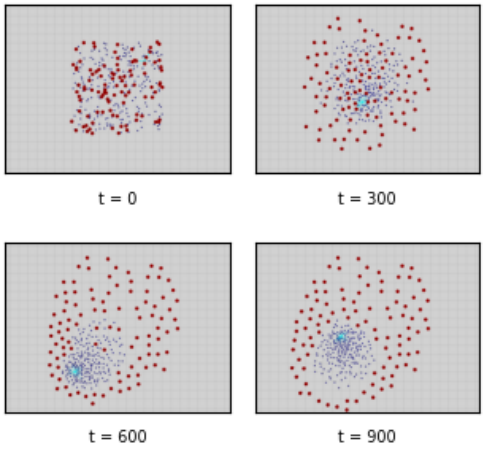

Figure 3: View of DBP-based target tracking in ARGoS. Byzantine robots are highlighted with red circles, direct observations of the target are shown in cyan.

Experiment result. In Fig. 4 we plot the belief that each cooperative robot has about the x-coordinate of the target, summarized using a quantile heatmap. The range of beliefs decreases until approx. $t = 400$, at which point all of the Byzantine robots have been blocked and the execution enters the regime with all Byzantine influence removed. Views of the DBP target tracking experiment in the ARGoS simulator are shown in Fig. 3. The baseline W-MSR algorithm requires the resilience parameter F to be picked a priori.

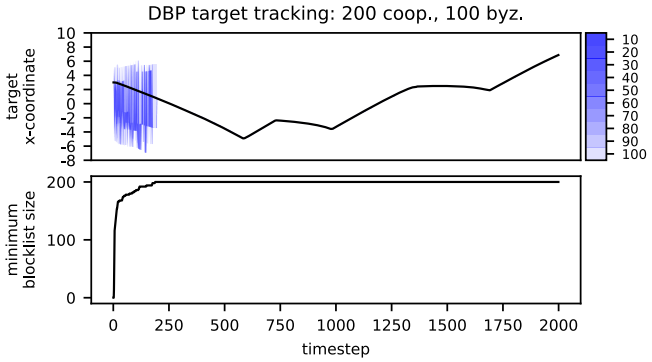

Figure 4: DBP-based target tracking performance. At top, the black curve shows the true x-coordinate of the moving target and the shaded blue regions show the range of beliefs as percentiles around the median. At bottom we plot $\min_i R_i^*[t]$, i.e. the minimum blocklist size. At timestep ~ 200, all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

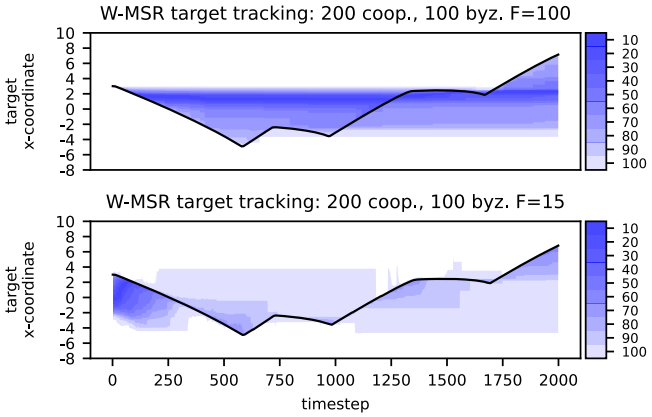

Figure 5: W-MSR-based target tracking performance. When the resilience parameter $F = 100$ (top) in order to guarantee safety, the information about the moving target cannot propagate through the cooperative robots due to the high connectivity and simultaneous observer requirements. As a comparison,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

If F is chosen too small, the theoretical guarantees of W-MSR are forfeited so the cooperative robots' consensus may be disrupted by the Byzantine robots. Specifically, part of the swarm where the density of Byzantine robots is low may be able to track the target successfully, however cooperative robots with more than F Byzantine neighbors will be affected by the attack. Each cooperative robot affected by the attack will in turn strengthen the attack as their local value nears the attacker's value – this scenario is shown with $F = 15 < |\bar{C}|$ at the bottom of Fig. 5. However, W-MSR does not scale to large F, since the robots cannot achieve such a high level of network connectivity and also high number of cooperative observers. The large-F regime is shown at top in Fig. 5, with $F = 100 = |\bar{C}|$.

5.2 Time Synchronization

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

(1) the observer's unique ID
(2) the observed time

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

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

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

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

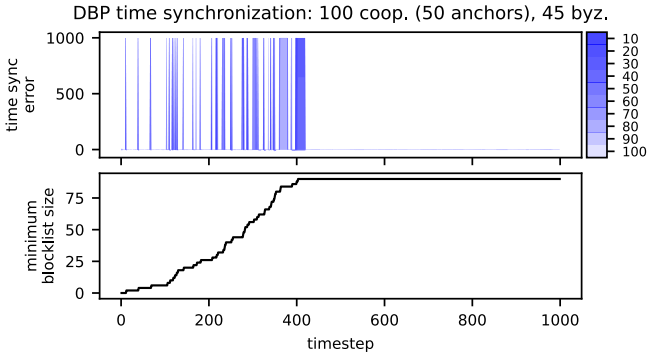

Figure 6: DBP-based time synchronization performance. Shown at top is the error between the cooperative robots' local clocks relative to the global reference clock. The error spikes to the attacker offset whenever a Byzantine robot initiates an attack. After timestep ~ 400, all of the Byzantines have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

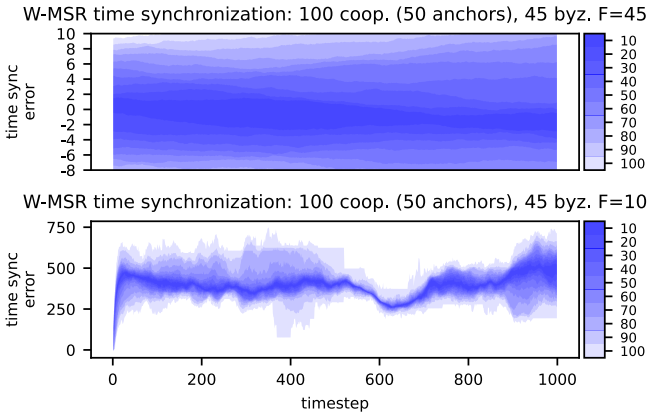

Figure 7: W-MSR-based time synchronization performance. As in the target tracking case study, $F = |\bar{C}| = 45$ guarantees safety, but the associated connectivity requirement prohibits convergence. Conversely, a lower $F = 10$ permits convergence of the consensus at the cost of allowing the Byzantines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

Experiment result. In Fig. 6 we plot the time synchronization error (difference between local time and the reference time) of the cooperative non-anchor robots over the course of the simulation. We observe that the Byzantine robots are able to push the synchronization error to the attack offset of +1000 timesteps by transmitting the false reference time observations, until timestep ~ 400, at which point each cooperative robot has blocked all of the Byzantine robots and the attacker influence is successfully removed. As with the target tracking case study, the W-MSR baseline requires a choice to be made a priori for the resilience parameter, F. If F is chosen too small, e.g. $F = 10$ shown at bottom in Fig. 7, then the Byzantine robots influence the consensus and the cooperative robots have local clock values between the attack offset and the reference time. If F is chosen large enough to be resilient to $|\bar{C}| = 45$ attackers, then the connectivity and simultaneous observation requirement is too large for the non-anchor robots to update their local clocks from neighbor's observations. The large-F regime is shown at top in Fig. 7 with $F = 45 = |\bar{C}|$.

5.3 Cooperative Localization

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

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

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

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

Accusation rules. Received localization messages are subjected to two accusation rules. The first rule is applied by anchor robots when receiving localization messages from other anchors, either directly or as attachments to non-anchor localization messages. Given that the other anchor j claims to be at $\tilde{p}_j$ at time s, let $\Delta t = t - s$ the elapsed time and $\Delta x_i = \|\tilde{p}_j - p_i\|$. The receiving anchor i will accuse j if $c\Delta t < \Delta x_i$, or in other words, if the anchor j's localization message has traveled faster-than-possible through the network. The second accusation rule can be issued by all robots, including non-anchors. The second rule asserts that the first rule

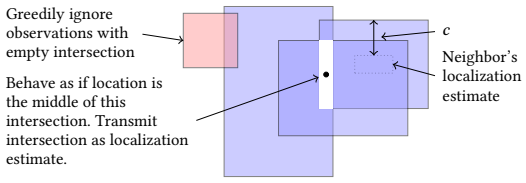

Figure 8: Observation-based cooperative localization setup for use with DBP. Non-anchor robots estimate their localization based on localization estimates received from their neighbors. Estimates are dilated by the transmission distance, and then reduced with the set intersection operator to compute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

hold between any received non-anchor localization message and its attached anchor message. These simple accusations could be extended if the robot capabilities were better. For example, if the robots could measure a lower bound on the distance from senders, anchors would be able to issue analogous accusations in situations where localization messages from other anchors should have been received sooner.

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

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

6 CONCLUSION

This work has proposed the use of a decentralized blocklist protocol based on inter-robot accusations as a means to provide Byzantine

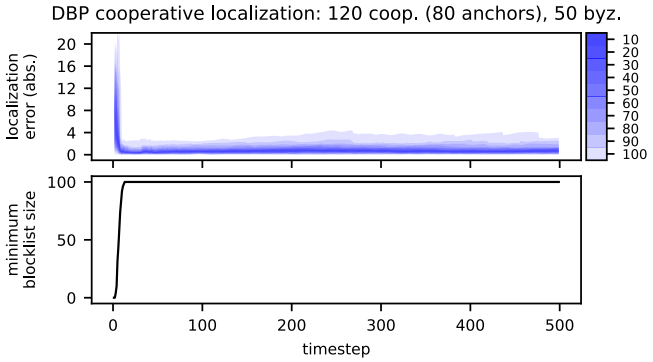

Figure 9: DBP-based cooperative localization performance. At top, we plot the absolute error that the cooperative robots have in the estimate of the x-coordinate of their position. At bottom we plot the minimum size of the cooperative robots' blocklists – once all of the Byzantines are blocked the estimation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

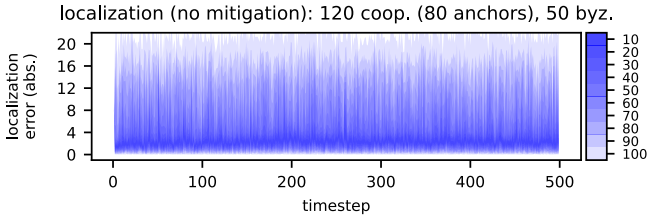

Figure 10: To support our claim that DBP is a suitable approach for this task, we show the impact that Byzantine robots can have on cooperative localization – Byzantines can cause the cooperative robots to have arbitrarily large localization errors.

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

7 ACKNOWLEDGEMENTS

We gratefully acknowledge the support from the National Science Foundation awards CNS-1932162, CNS-1931997, and GRFP-1938052.

REFERENCES

[1] USP Srinivas Aditya, Roshan Singh, Pranav Kumar Singh, and Anshuman Kalla. 2021. A Survey on Blockchain in Robotics: Issues, Opportunities, Challenges and Future Directions. *Journal of Network and Computer Applications* 196 (2021), 103245.

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

[3] Filippo Arrichiello, Alessandro Marino, and Francesco Pierri. 2015. Observer-based decentralized fault detection and isolation strategy for networked multirobot systems. *IEEE Transactions on Control Systems Technology* 23, 4 (2015), 1465–1476. Publisher: IEEE.

[4] Yotam Ashkenazi, Shlomi Dolev, Sayaka Kamei, Fukuhito Ooshita, and Koichi Wada. 2019. Forgive & Forget: Self-Stabilizing Swarms in Spite of Byzantine Robots. In *2019 Seventh International Symposium on Computing and Networking Workshops (CANDARW)*. 188–194. https://doi.org/10.1109/CANDARW.2019.00041

[5] Shahriar Bijani and David Robertson. 2014. A review of attacks and security approaches in open multi-agent systems. *Artificial Intelligence Review* 42, 4 (2014), 607–636. Publisher: Springer.

[6] Mingxi Cheng, Chenzhong Yin, Junyao Zhang, Shahin Nazarian, Jyotirmoy Deshmukh, and Paul Bogdan. 2021. A general trust framework for multi-agent systems. In *Proceedings of the 20th International Conference on Autonomous Agents and MultiAgent Systems*. 332–340.

[7] Seddik M Djouadi, Alexander M Melin, Erik M Ferragut, Jason A Laska, Jin Dong, and Anis Drira. 2015. Finite energy and bounded actuator attacks on cyber-physical systems. In *2015 European Control Conference (ECC)*. IEEE, 3659–3664.

[8] Jack Edmonds. 1965. Paths, trees, and flowers. *Canadian Journal of mathematics* 17 (1965), 449–467.

[9] Adriano Fagiolini, Marco Pellinacci, Gianni Valenti, Gianluca Dini, and Antonio Bicchi. 2008. Consensus-based Distributed Intrusion Detection for Multi-Robot Systems. In *2008 IEEE International Conference on Robotics and Automation*. 120–127. https://doi.org/10.1109/ROBOT.2008.4543196 ISSN: 1050-4729.

[10] Adriano Fagiolini, Gianni Valenti, Lucia Pallottino, Gianluca Dini, and Antonio Bicchi. 2007. Decentralized intrusion detection for secure cooperative multi-agent systems. In *2007 46th IEEE Conference on Decision and Control*. IEEE, 1553–1558.

[11] Jennifer Gielis, Ajay Shankar, and Amanda Prorok. 2022. A Critical Review of Communications in Multi-robot Systems. *Current Robotics Reports* (Aug. 2022). https://doi.org/10.1007/s43154-022-00090-9

[12] Stephanie Gil, Swarun Kumar, Mark Mazumder, Dina Katabi, and Daniela Rus. 2017. Guaranteeing spoof-resilient multi-robot networks. *Autonomous Robots* 41, 6 (2017), 1383–1400. Publisher: Springer.

[13] Luis Guerrero-Bonilla, Amanda Prorok, and Vijay Kumar. 2017. Formations for Resilient Robot Teams. *IEEE Robotics and Automation Letters* 2, 2 (April 2017), 841–848. https://doi.org/10.1109/LRA.2017.2654550 Conference Name: IEEE Robotics and Automation Letters.

[14] Pinyao Guo, Hunmin Kim, Nurali Virani, Jun Xu, Minghui Zhu, and Peng Liu. 2018. RoboADS: Anomaly detection against sensor and actuator misbehaviors in mobile robots. In *2018 48th Annual IEEE/IFIP international conference on dependable systems and networks (DSN)*. IEEE, 574–585.

[15] Heath J. LeBlanc, Haotian Zhang, Xenofon Koutsoukos, and Shreyas Sundaram. 2013. Resilient Asymptotic Consensus in Robust Networks. *IEEE Journal on Selected Areas in Communications* 31, 4 (April 2013), 766–781. https://doi.org/10.1109/JSAC.2013.130413 Conference Name: IEEE Journal on Selected Areas in Communications.

[16] Jun Liu, Lifeng Zhou, Pratap Tokekar, and Ryan K. Williams. 2021. Distributed Resilient Submodular Action Selection in Adversarial Environments. *IEEE Robotics and Automation Letters* 6, 3 (July 2021), 5832–5839. https://doi.org/10.1109/LRA.2021.3080629 arXiv: 2105.07305.

[17] Frederik Mallmann-Trenn, Matthew Cavorsi, and Stephanie Gil. 2021. Crowd Vetting: Rejecting Adversaries via Collaboration With Application to Multirobot Flocking. *IEEE Transactions on Robotics* (2021), 1–20. https://doi.org/10.1109/TRO.2021.3089033 Conference Name: IEEE Transactions on Robotics.

[18] Aritra Mitra, John A. Richards, Saurabh Bagchi, and Shreyas Sundaram. 2019. Resilient distributed state estimation with mobile agents: overcoming Byzantine adversaries, communication losses, and intermittent measurements. *Autonomous Robots* 43, 3 (March 2019), 743–768. https://doi.org/10.1007/s10514-018-9813-7

[19] Alexandre Pacheco, Volker Strobel, and Marco Dorigo. 2020. A blockchain-controlled physical robot swarm communicating via an ad-hoc network. In *International Conference on Swarm Intelligence*. Springer, 3–15.

[20] Alyssa Pierson and Mac Schwager. 2016. Adaptive Inter-Robot Trust for Robust Multi-Robot Sensor Coverage. In *Robotics Research*, Masayuki Inaba and Peter Corke (Eds.). Vol. 114. Springer International Publishing, Cham, 167–183. https://doi.org/10.1007/978-3-319-28872-7_10 Series Title: Springer Tracts in Advanced Robotics.

[21] Carlo Pinciroli, Vito Trianni, Rehan O'Grady, Giovanni Pini, Arne Brutschy, Manuele Brambilla, Nithin Mathews, Eliseo Ferrante, Gianni Di Caro, Frederick Ducatelle, Mauro Birattari, Luca Maria Gambardella, and Marco Dorigo. 2012. ARGoS: a Modular, Parallel, Multi-Engine Simulator for Multi-Robot Systems. *Swarm Intelligence* 6, 4 (2012), 271–295.

[22] Venkatraman Renganathan and Tyler Summers. 2017. Spoof resilient coordination for distributed multi-robot systems. In *2017 International Symposium on Multi-Robot and Multi-Agent Systems (MRS)*. IEEE, 135–141.

[23] David Saldaña, Amanda Prorok, Shreyas Sundaram, Mario F. M. Campos, and Vijay Kumar. 2017. Resilient consensus for time-varying networks of dynamic agents. In *2017 American Control Conference (ACC)*. 252–258. https://doi.org/10.23919/ACC.2017.7962962 ISSN: 2378-5861.

[24] Kelsey Saulnier, David Saldaña, Amanda Prorok, George J. Pappas, and Vijay Kumar. 2017. Resilient Flocking for Mobile Robot Teams. *IEEE Robotics and Automation Letters* 2, 2 (April 2017), 1039–1046. https://doi.org/10.1109/LRA.2017.2655142 Conference Name: IEEE Robotics and Automation Letters.

[25] Yasser Shoukry, Shaunak Mishra, Zutian Luo, and Suhas Diggavi. 2018. Sybil attack resilient traffic networks: A physics-based trust propagation approach. In *2018 ACM/IEEE 9th International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 43–54.

[26] Volker Strobel, Eduardo Castelló Ferrer, and Marco Dorigo. 2018. Managing Byzantine Robots via Blockchain Technology in a Swarm Robotics Collective Decision Making Scenario. In *Proceedings of the 17th International Conference on Autonomous Agents and MultiAgent Systems* (Stockholm, Sweden) *(AAMAS '18)*. International Foundation for Autonomous Agents and Multiagent Systems, Richland, SC, 541–549.

[27] Janardan Kumar Verma and Virender Ranga. 2021. Multi-Robot Coordination Analysis, Taxonomy, Challenges and Future Scope. *Journal of Intelligent & Robotic Systems* 102, 1 (May 2021), 10. https://doi.org/10.1007/s10846-021-01378-2

[28] Matthew Weber, Baihong Jin, Gil Lederman, Yasser Shoukry, Edward A. Lee, Sanjit Seshia, and Alberto Sangiovanni-Vincentelli. 2020. Gordian: Formal Reasoning-based Outlier Detection for Secure Localization. *ACM Transactions on Cyber-Physical Systems* 4, 4 (Aug. 2020), 1–27. https://doi.org/10.1145/3386568

[29] Jean-Paul A. Yaacoub, Hassan N. Noura, Ola Salman, and Ali Chehab. 2022. Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. *International Journal of Information Security* 21, 1 (Feb. 2022), 115–158. https://doi.org/10.1007/s10207-021-00545-8

[30] Lifeng Zhou and Pratap Tokekar. 2021. Multi-Robot Coordination and Planning in Uncertain and Adversarial Environments. *Current Robotics Reports* 2, 2 (June 2021), 147–157. https://doi.org/10.1007/s43154-021-00046-5 arXiv: 2105.00389.

[31] Lifeng Zhou, Vasileios Tzoumas, George J. Pappas, and Pratap Tokekar. 2020. Distributed Attack-Robust Submodular Maximization for Multi-Robot Planning. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*. 2479–2485. https://doi.org/10.1109/ICRA40945.2020.9197243 ISSN: 2577-087X.