Leveraging Blockchain for Spoof-Resilient Robot Networks

Tauhidul Alam[1(✉)], Jarrett Taylor[1], Jonathan Taylor[1], Shahriar Badsha[2], Abdur Rahman Bin Shahid[3], and A.S.M. Kayes[4]

[1] Department of Computer Science, Louisiana State University Shreveport, Shreveport, LA 71115, USA
{talam,taylorj48,taylorj6975}@lsus.edu
[2] Department of Computer Science and Engineering, University of Nevada, Reno, NV 89557, USA
sbadsha@unr.edu
[3] Department of Computer Science, Concord University, Athens, WV 24712, USA
ashahid@concord.edu
[4] Department of Computer Science and Information Technology, La Trobe University, Bundoora, Victoria 3086, Australia
a.kayes@latrobe.edu.au

Abstract. Autonomous robots, such as unmanned aerial or ground robots, are vulnerable to cyber attacks since they use sensor data heavily for their path planning and control. Furthermore, consensus is critical for resilient coordination and communication of robots in multi-robot networks against a specific adversarial attack called the spoofing attack, where robots can be compromised by an adversary. Therefore, we leverage Blockchain in a network of robots to coordinate their path planning and present a consensus method utilizing their transferred Blockchain data to detect compromised robots. Our simulation results corroborate the fact that the proposed method enhances the resilience of a robot network by detecting its spoofed client robots or compromised server at a significant rate during the spoofing attack.

Keywords: Robot networks · Spoofing attack · Blockchain · Communication

1 Introduction

Robot networks have made a notable impact in several applications such as drone delivery, infrastructure inspections, disaster information gathering, agriculture precision, border and area surveillance, and search and rescue operations. An example application of robot networks is Wing's drones [1] certified by the Federal Aviation Administration (FAA) in the U.S. for the first time that deliver small packages, including food, medicine, and household items, directly to homes in minutes following flight paths. In practice, these robots in a network utilize wireless communication sensors for their path planning, coordination, control,

C. S. Chan et al. (Eds.): ICIRA 2020, LNAI 12595, pp. 207–216, 2020.
https://doi.org/10.1007/978-3-030-66645-3_18

and collision avoidance. However, malicious agents can jam or intercept these wireless sensors to gain access to the network that makes the robots vulnerable to cyber attacks and malicious traffic. In particular, a robot network can be disrupted by a spoofing attack which is also known as a Sybil attack [4]. An adversary can forge multiple spurious identities or impersonate several existing client robots in the network during the spoofing attack [13] after having complete control over GPS [16] or optical flow sensors [3].

Our work is motivated by the problem of defending a robot network against the spoofing attack, e.g.., a set of client robots is drawn away by an adversary from their service robot as illustrated in Fig. 1. Specifically, we consider a robot network in which a group of aerial delivery vehicles delivers packages launching from a central fulfillment or distribution station to designated customer (goal) locations and a server robot controls the operations of delivery vehicles. However, an adversary attempts to gain control of multiple delivery vehicles by spoofing their customer locations or compromising the server robot. This spoofing attack is easy to carry out but difficult to prevent in multi-robot settings. Consequently, our problem of safeguarding the network by detecting this attack is challenging.

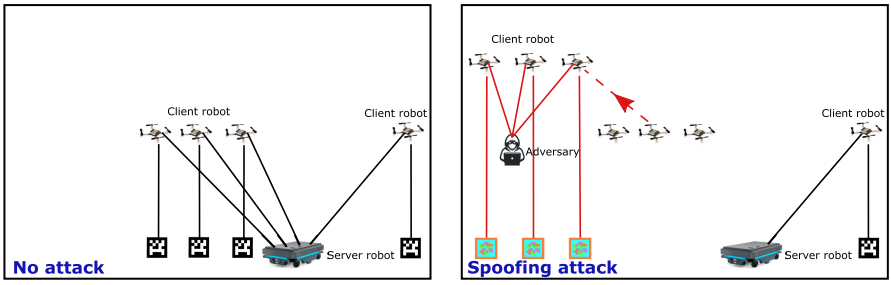

Fig. 1. Spoofing attack. A server robot controls pre-computed flight paths to client robots for their intended customer locations when no attack is present. In a spoofing attack, an adversary spoofs many client robots by drawing them away from their original paths and directing them toward pseudo-customer locations.

Our work is closely related to spoof-resilient solutions to multi-robot networks using information extracted from Wi-Fi communication signals for detecting spoofed client robots [7] and providing a consensus algorithm with bounded performance guarantees [6]. Consensus methods in mobile and distributed networks are also considered using transmitted values [14,15] to remove adversarial agents from the consensus and using exchanged keys [8] or tokens [9] to secure networks with different initial topologies [20]. However, these consensus methods are prone to failures when robots in a network fail or communicate incorrect messages. Additionally, a malicious agent can generate a number of false identities in a robot network that utilizes wireless signals for security instead of using a trusted system within the network. Unlike prior work, this method utilizes trusted Blockchain technology for the resilient coordination and communication

of other robots in the network in the presence of malfunctioning and malicious robots (Byzantine robots).

The potential benefits of using Blockchain technology for addressing security issues in swarm robotic systems are discussed in [5]. The Nakamoto's white paper [10] first introduced Blockchain technology as a trusted database of encrypted and linked data transactions with timestamps stored by participating agents of a peer-to-peer network. There are very few approaches that exploit Blockchain technology for robotic security systems. A Blockchain-based collective decision making approach [17,19] for managing Byzantine robots in homogeneous robot swarms is presented. Blockchain has been utilized in heterogeneous robot swarms as well for collaboration in [12]. Existing consensus algorithms are compared with the Blockchain consensus algorithm in [18]. The pivotal advantages of using Blockchain are the immutability of transactions, decentralized consensus, fault tolerance, and so on. As such, this work leverages a permissioned or private Blockchain, where a centralized entity has control over its participants, to detect spoofed client robots or the compromised server through the validation by a devised committee of robots in the network.

Contributions: This paper makes the following contributions.

- A consensus method with a subset of robots in the committee making use of transferred data transactions on Blockchain for detecting compromised robots in the network.
- A simulation study that validates the performance of our method for different types of compromised robots in dealing with the spoofing attack.

The remainder of this paper is laid out as follows. First, we define several notations required for our robot network setting and formulate our problem of interest in Sect. 2. Then, we describe our method to detect the spoofing attack in the network in Sect. 3. The results from the implementation of our method appear next in Sect. 4. Finally, we summarize our paper along with future directions in Sect. 5.

2 Preliminaries

We examine a robot network setting in which m client delivery robots (drones) $\mathcal{D} = \{D_1, \ldots, D_m\}$ obtain computed flight paths $\mathcal{T} = \{\tau_1, \ldots, \tau_m\}$ for their product delivery from a server robot S located at a central distribution center or service station. These client delivery robots are identified by a set of identification keys which are denoted as $\mathcal{I} = \{i_1, \ldots, i_m\}$. They communicate their path information (location, velocity, time, distance, etc.) with the server robot through the identification keys $\mathcal{I}$. Let $P = \{p_1, \ldots, p_m\}$ denote the client delivery robots' locations in $\mathbb{R}^3$. Let $V = \{v_1, \ldots, v_m\}$ denote the client delivery robots' velocities in $\mathbb{R}$. Let $G = \{g_1, \ldots, g_m\}$ denote the client delivery robots' goal or customer locations in $\mathbb{R}^2$. Let p_s be the location of S in $\mathbb{R}^2$. Let a flight path of a client delivery robot D_k, where $k \in \{1, \ldots, m\}$, be $\tau_k : [0, t] \rightarrow \mathbb{R}^3$ such

that $\tau_k(0) = p_s$ and $\tau_k(t) = g_k$ for a finite time interval $[0, t]$. We consider in this setting that either a subset of client delivery robots denoted by $\mathcal{A}$, where $\mathcal{A} \subset \mathcal{D}$, can be spoofed, or the server S is compromised by adversaries. It is assumed that an adversary sends various messages over the network with identification keys to make client delivery robots spoofed by taking control over their GPS sensors and that the knowledge of which client delivery robots are spoofed is unknown [6]. Furthermore, all the client delivery robots are considered to be spoofed when the server robot is compromised. In this context, we formulate the following problem of interest.

Problem 1 (*Detecting spoofed robots*): *Given m client delivery robots, the server robot S in a network, and their pre-computed paths $\mathcal{T}$, detect $\mathcal{A}$ spoofed client delivery robots or the compromised server S in the case of a spoofing attack.*

3 Methods

This section details our Blockchain leveraged consensus method for detecting the spoofing attack in a robot network.

In our method, we first construct a server robot, and then it establishes a network with m client delivery robots. The server robot provides the identification keys $\mathcal{I}$ to client delivery robots for communication and sends their pre-computed paths $\mathcal{T}$ toward their goal locations G.

In our next step, we employ a private Blockchain on the server robot in order to keep track of transferred data over the network. Client delivery robots communicate data related to their locations, velocities, distances, and time periodically with the server robot, and the server robot incorporates them into the transferred Blockchain data. The transferred Blockchain data is defined as B. We assume that client delivery robots act honestly in communicating their data.

Afterward, we develop a consensus Algorithm 1 in the robot network for detecting the spoofing attack. To achieve this, we devise a verification committee with the server robot and a subset of random client delivery robots. The server robot alone can also detect spoofed client delivery robots but cannot detect itself being compromised. Let $\mathcal{C} \subset \mathcal{D} \setminus \mathcal{A}$ be the subset of client delivery robots in the verification committee. It is considered that n client committee members, where $|\mathcal{C}| = n$ and $n < m$, can access the transferred Blockchain data B. We also consider that both the server and the client committee members are not compromised at the same time.

The verification committee members can vote in a weighted manner for detecting spoofed client delivery robots or a compromised server. Let w_c be the weight for each committee member, including the server robot. Let w_s be the weight for the server robot's vote and w_{cc} be the weight for each client committee robot's vote. The client committee robot's weight for voting is calculated as $w_{cc} = (1 - w_s)/n$. The weights for all committee members can be represented as a $(n + 1)$-dimensional vector as follows.

$$w = (w_1, \ldots, w_{n+1}) = (w_s, w_1, \ldots, w_n).$$

The elements of w should satisfy two conditions: 1) $w_c > 0$ for all $c \in \{1, \ldots, n+1\}$ and 2)

$$\sum_{c=1}^{n+1} w_c = w_s + \sum_{cc \in \mathcal{C}} w_{cc} = 1.$$

Let Q be the consensus trigger by the first committee member to start the consensus process. Let $\mathcal{L} = \{l_1, \ldots, l_m\}$ be the set of each committee member's votes for client delivery robots. Let π be the threshold value for the path deviation of a client delivery robot. In Algorithm 1, we calculate the set of votes for client delivery robots to determine one or more spoofed client delivery robots. We apply the consensus Algorithm 1 for each verification committee member, including the server robot. For each client delivery robot with an identification key i_k, we check its path deviation from the provided path τ_k by the server robot. Since we consider that a client delivery robot's GPS is spoofed, we make use of its locations data that are stored in Blockchain to find the path deviation. Thus, the PathDeviation function takes the input of Blockchain data B, identification key i_k, and path τ_k with the location of a client delivery robot's original destination. The function iterates through B to compare the client delivery robot's current location to its goal location to determine if the client delivery robot is getting closer to its intended destination. Once determined, the function returns a value t between 0 and 1. If it is less than the threshold value π , the algorithm determines that the client delivery robot is deviating from its intended path. Then, we add $-w_c$ to the set of votes for that client delivery robot. Otherwise, we add w_c to the set for the same client delivery robot.

Algorithm 1: Consensus $(Q, \mathcal{T}, \mathcal{I}, B, w_c)$

Input: $Q, \mathcal{T}, \mathcal{I}, B, w_c$ – Consensus trigger, Client delivery robots' paths, Set of identification keys for client delivery robots, Blockchain data, Weight of each committee member

Output: $\mathcal{L}$ – Set of votes for client delivery robots

```
for k =1 to |I| do
    t ← PathDeviation(B, i_k, τ_k)
    if t < π then
        L ← L ∪ {−w_c}
    else
        L ← L ∪ {w_c}
return L
```

In our final step, we investigate two spoofing attack scenarios: 1) client delivery robots are spoofed and 2) the server is compromised. The verification committee validates these attack scenarios utilizing the transferred Blockchain data

B by detecting the compromised server or spoofed client delivery robots. In both scenarios, one of the client committee members begins the consensus by voting for each client delivery robot and alerting the other committee members, including the server, to initiate their voting. Therefore, the rest of the committee members also provide their votes for each client delivery robot. Once a consensus is reached by all the committee members, the total votes are combined to a tally for each client delivery robot. If the vote tally for a client delivery robot less than zero, the associated client delivery robot is detected as spoofed.

For the first scenario, a set of client delivery robots $\mathcal{D}$ is launched with n client committee members (robots) by the server. A subset of client delivery robots $\mathcal{A}$ is spoofed and changes directions mid-way to their intended destinations. The committee members detect a spoofed client delivery robot through their consensus algorithm after it moves away from its intended destination. It is important to mention for this scenario that the server robot itself can also detect a spoofed client delivery robot without the verification of other client committee members as both the server and client committee members can utilize the transferred Blockchain data B for verification.

For the second scenario, the server is no longer considered as a committee member. The n client committee members are relied upon to complete the consensus using their accessible communicated Blockchain data B. Once the consensus is initiated and completed, the votes are tallied and checked. All the client delivery robots will be detected as spoofed because the compromised server automatically spoofs non-committee client members. Once all non-committee client members are detected as spoofed, the server will be detected as compromised.

4 Experimental Results

In this section, we present the results from the implementation of our method.

Initially, we implemented a robot network through server-client socket programming in Python 3 with the simulation of port numbers and identification keys for robots. The server robot provided flight paths to client delivery robots (drones). Then, we simulated our own private Blockchain in this network setting using Python. Client delivery drones followed the provided flight paths which were simulated taking advantage of a Python Robotics tool [2]. While client delivery drones following the simulated paths, they communicated their locations, velocities, covered distances, and time with the server robot. These communicated data were transferred over the network through our implemented Blockchain.

We also implemented our consensus Algorithm 1 in simulation. In our implementation, we accounted for $n = 2$ random client delivery drones and the server robot for devising the verification committee. These committee members employed transferred Blockchain data for validation of non-committee client delivery drones' path deviation using their locations along their flight paths. For the voting process of the verification committee members, the weights we utilized for the server robot and the client delivery drone are $w_s = 0.4$ and $w_{cc} = 0.3$

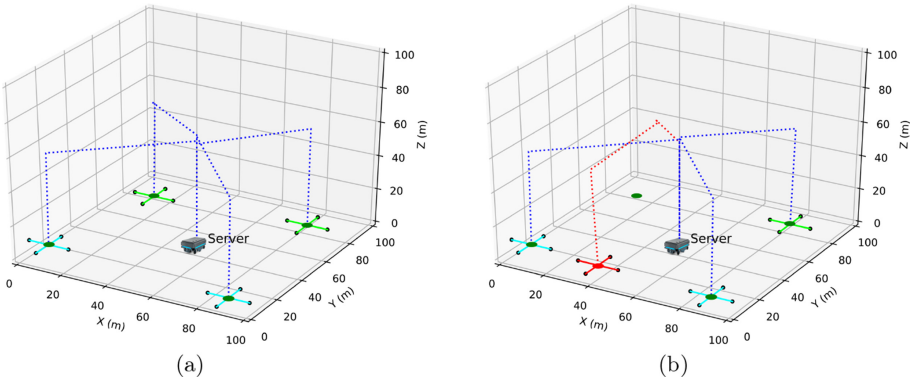

Fig. 2. Client delivery drone spoofing scenario. Flight paths of two client committee drones (depicted in cyan) toward their desired goal locations (green circles) and one non-committee drones (depicted in green) toward its desired goal location and another (depicted in red) toward its unintended goal location (red circle) starting from the server robot's location. Between two non-committee drones, the red non-committee drone is spoofed and detected. (Color figure online)

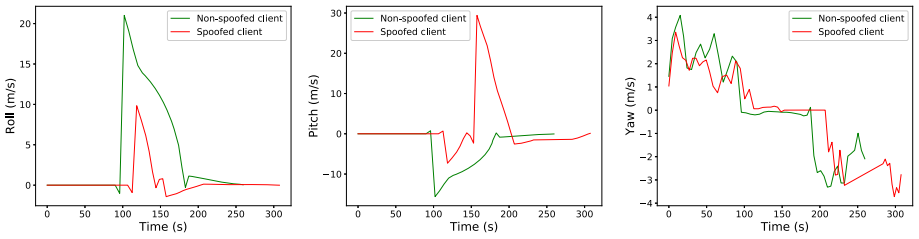

Fig. 3. Comparison of velocities and time for a non-spoofed client delivery drone and its spoofed counterpart after detection.

respectively. The threshold value $\pi = 0.9$ was used for finding the path deviation of a client delivery drone. Finally, the verification committee members recorded their votes for detecting spoofed client delivery drones.

Figure 2(a) delineates a drone delivery network setting, where $m = 4$ client delivery drones were launched from the server robot's location toward their green goal locations with $n = 2$ client committee drones marked in cyan and the remaining client delivery drones marked in green. Figure 2(b) presents the first spoofing attack scenario, where one of the non-committee client drones was attacked during its flight path execution and moved away from its intended goal location. In our implementation, the verification committee members detected the client delivery drone as spoofed based on their votes and turned it red, including its path and spoofed goal location. Figure 3 illustrates the variations of distinct velocities (roll, yaw, pitch) with respect to time for a non-spoofed client delivery drone and its detected spoofed counterpart. These results indi-

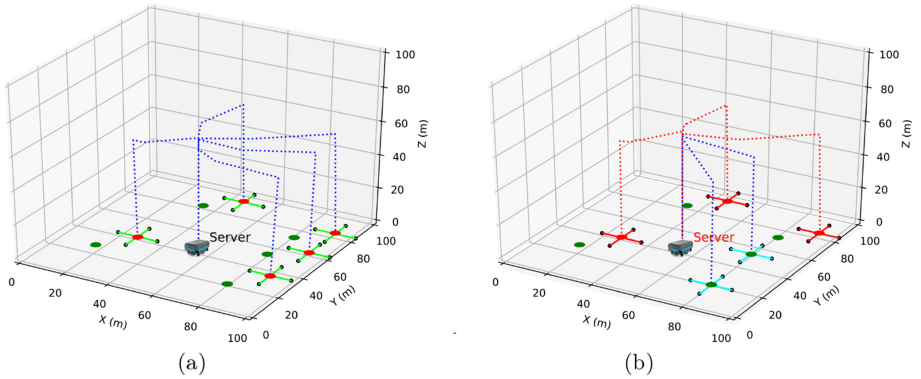

Fig. 4. Server robot compromising scenario. (a) Flight paths of five client delivery drones (depicted in green) starting from the server robot's location toward their spoofed goal locations (red circles) when the server robot is compromised and undetected. (b) Flight paths of two client committee drones (depicted in cyan) toward their desired goal locations (green circles) and three non-committee drones (depicted in red) toward their spoofed goal locations (red circles) starting from the server robot's location when it is detected that the server robot is compromised. (Color figure online)

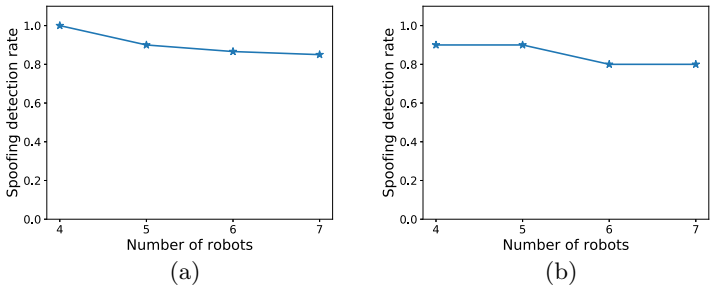

Fig. 5. Spoofing detection rate of our method for different numbers of robots in a network for the first scenario (a) and the second scenario (b).

cate the inconsistencies in velocities between an actual flight path and a deviated flight path of a client delivery drone when it is attacked.

Figure 4(a) shows a spoofing attack scenario where $m = 5$ drones were launched; however, the server was compromised and could not run its own consensus nor could any of the five drones launch their own consensus. Since the server was compromised, it redirected all non-committee drones to spoofed goal locations. Since no committee drones were present, this resulted in no drones being detected as spoofed. Figure 4(b) demonstrates the results of the second spoofing attack scenario, where five client delivery drones were launched but later the server was compromised. In this case, we converted $n = 2$ client delivery drones into committee members. As a result, the committee members were able to detect the remaining client delivery drones as spoofed. Since all non-

committee drones were spoofed, it detected that the server was compromised which was depicted by coloring its name to red.

We computed the spoofing detection rate of our method for different numbers of robots, including the server robot, in a network for both scenarios which is illustrated in Fig. 5. The spoofing detection rate was computed from the average of 10 runs of our implementation for each number of robots for both scenarios. This result shows that our detection rate is significant but decreases slightly with the increase in the number of robots. The reason for this small detection rate decline is that the client committee members sometimes complete their flights or do not even start their flights while some non-committee client members are spoofed. This problem can be overcome by dynamically assigning client committee members that are on their flights to the verification committee.

5 Conclusion and Future Directions

In this paper, we presented a consensus method with a committee of robots in a network for detecting its spoofed client robots or compromised server utilizing transferred Blockchain data. Our simulation results demonstrate that our method makes a robot network resilient against the spoofing attack. We believe that we have just scratched the surface in leveraging Blockchain for detecting a cyber attack within a robot network. This effort paves the way for several interesting future research directions as detailed below.

In the future efforts of this stream of research, we will evaluate the vulnerabilities of our method by learning the characteristics of compromised robots by different attacks on a secure network using machine learning methods, and present solutions to these vulnerabilities to make our method more attack resilient. We also plan to test our method with a set of programmable drones as client robots and a ground vehicle as the server robot.

One potential problem of our method lies in storing transferred Blockchain data from a group of robots in a network due to the increase of its storage while the network keeps running with a large number of robots. To alleviate this problem, we will investigate an approach to reduce Blockchain data by transferring them on-demand or storing only hash values for these data [11].

Acknowledgements. This work is supported in part by the Louisiana Board of Regents Contract Number LEQSF(2020-21)-RD-A-14.

References

1. Wing - A commercial drone delivery service. https://wing.com. Accessed 30 June 2020
2. Python Robotics. https://github.com/AtsushiSakai/PythonRobotics. Accessed 5 April 2020
3. Davidson, D., Wu, H., Jellinek, R., Singh, V., Ristenpart, T.: Controlling UAVs with sensor input spoofing attacks. In: Proceedings of the 10th USENIX Workshop on Offensive Technologies (2016)

4. Douceur, J.R.: The sybil attack. In: Druschel, P., Kaashoek, F., Rowstron, A. (eds.) IPTPS 2002. LNCS, vol. 2429, pp. 251–260. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45748-8_24
5. Castelló Ferrer, E.: The blockchain: a new framework for robotic swarm systems. In: Arai, K., Bhatia, R., Kapoor, S. (eds.) FTC 2018. AISC, vol. 881, pp. 1037–1058. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-02683-7_77
6. Gil, S., Baykal, C., Rus, D.: Resilient multi-agent consensus using Wi-Fi signals. IEEE Control Syst. Lett. **3**(1), 126–131 (2018)
7. Gil, S., Kumar, S., Mazumder, M., Katabi, D., Rus, D.: Guaranteeing spoof-resilient multi-robot networks. Auton. Robots **41**(6), 1383–1400 (2017). https://doi.org/10.1007/s10514-017-9621-5
8. LeBlanc, H.J., Zhang, H., Koutsoukos, X., Sundaram, S.: Resilient asymptotic consensus in robust networks. IEEE J. Sel. Areas Commun. **31**(4), 766–781 (2013)
9. Ma, H., Hönig, W., Kumar, T.S., Ayanian, N., Koenig, S.: Lifelong path planning with kinematic constraints for multi-agent pickup and delivery. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 33, pp. 7651–7658 (2019)
10. Nakamoto, S.: Bitcoin: a peer-to-peer electronic cash system. Technical Report (2019)
11. Nishida, Y., Kaneko, K., Sharma, S., Sakurai, K.: Suppressing chain size of blockchain-based information sharing for swarm robotic systems. In: Proceedings of International Symposium on Computing and Networking Workshops, pp. 524–528 (2018)
12. Queralta, J.P., Westerlund, T.: Blockchain-powered collaboration in heterogeneous swarms of robots. Front. Robot. AI (2020)
13. Renganathan, V., Summers, T.: Spoof resilient coordination for distributed multi-robot systems. In: Proceedings of the International Symposium on Multi-Robot and Multi-Agent Systems, pp. 135–141 (2017)
14. Sargeant, I., Tomlinson, A.: Modelling malicious entities in a robotic swarm. In: Proceedings of IEEE/AIAA Digital Avionics Systems Conference, pp. 7B1-1-7B1-12 (2013)
15. Saulnier, K., Saldana, D., Prorok, A., Pappas, G.J., Kumar, V.: Resilient flocking for mobile robot teams. IEEE Robot. Autom. lett. **2**(2), 1039–1046 (2017)
16. Shepard, D.P., Bhatti, J.A., Humphreys, T.E., Fansler, A.A.: Evaluation of smart grid and civilian UAV vulnerability to GPS spoofing attacks. In: Proceedings of Radionavigation Laboratory Conference (2012)
17. Strobel, V., Castelló Ferrer, E., Dorigo, M.: Managing byzantine robots via blockchain technology in a swarm robotics collective decision making scenario. In: Proceedings of the International Conference on Autonomous Agents and MultiAgent Systems, pp. 541–549 (2018)
18. Strobel, V., Castelló Ferrer, E., Dorigo, M.: Blockchain technology secures robot swarms: a comparison of consensus protocols and their resilience to byzantine robots. Front. Robot. AI **7**, 54 (2020)
19. Strobel, V., Dorigo, M.: Blockchain technology for robot swarms: a shared knowledge and reputation management system for collective estimation. In: Proceedings of International Conference on Swarm Intelligence, pp. 425–426 (2018)
20. Wheeler, T., Bharathi, E., Gil, S.: Switching topology for resilient consensus using Wi-Fi signals. In: Proceedings of the International Conference on Robotics and Automation, pp. 2018–2024 (2019)