Poster: May the Swarm Be With You: Sensor Spoofing Attacks Against Drone Swarms

Yingao (Elaine) Yao
The University of British Columbia
Vancouver, Canada
elainey@ece.ubc.ca

Pritam Dash
The University of British Columbia
Vancouver, Canada
pdash@ece.ubc.ca

Karthik Pattabiraman
The University of British Columbia
Vancouver, Canada
karthikp@ece.ubc.ca

ABSTRACT

Swarm robotics, particularly drone swarms, are used in various safety-critical tasks. While a lot of attention has been paid to improving swarm control algorithms for improved intelligence, the security implications of various design choices in swarm control algorithms have not been studied. We highlight how an attacker can exploit the vulnerabilities in swarm control algorithms to disrupt drone swarms. Specifically, we show that the attacker can target one swarm member (target drone) through sensor spoofing attacks, and indirectly cause *other* swarm members (victim drones) to veer off from their course, and potentially resulting in a crash. Our attack cannot be prevented by traditional software security techniques, and it is stealthy in nature as it causes seemingly benign deviations in drone swarms. Our initial results show that spoofing the position of a target drone by 5m is sufficient to cause other drones to crash into a front obstacle. Overall, our attack achieves 76.67% and 93.33% success rate with 5m and 10m spoofing deviation respectively.

CCS CONCEPTS

• **Security and privacy → Software and application security**; • **Computer systems organization → Embedded and cyber-physical systems**.

KEYWORDS

cyber physical systems; swarm robotics security; attacks

ACM Reference Format:
Yingao (Elaine) Yao, Pritam Dash, and Karthik Pattabiraman. 2022. Poster: May the Swarm Be With You: Sensor Spoofing Attacks Against Drone Swarms. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22), November 7–11, 2022, Los Angeles, CA, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3548606.3563535

1 INTRODUCTION

Drone swarms are a type of distributed cyber-physical system inspired by swarm intelligence. They consist of multiple drones that can communicate with each other. Drone swarms use swarm control algorithms to collaboratively accomplish the mission. They can carry out large-scale missions that cannot be performed by a single drone. For example, they are used in various applications such as logistics, surveillance, and search and rescue [2].

With the increasing adoption in real-world applications, drone swarms have been shown to be vulnerable to threats such as logic flaws [5] in swarm control algorithms and authentication attacks [4]. However, attacks exploiting such threats incur high costs (e.g., introducing an external attack drone [5]), and can be thwarted using techniques proposed in prior work [5, 6]. On the other hand, physical attacks [9] in drone swarms have not received much attention. These are attacks that feed the drone with erroneous sensor measurements via physical channels. Drones rely on these sensor measurements for correct operations, and such erroneous values will lead to the drone malfunctioning [3]. For instance, a GPS spoofing attack [9] sends malicious GPS signals to the victim drone, causing the drone to deviate from its mission.

In this paper, we propose a type of physical attack in drone swarms, called *faulty sensor propagation attacks*, that exploits sensor spoofing to *indirectly* cause disruptions in drone swarms. We design faulty sensor propagation attacks based on two observations we made. First, swarm control algorithms, the key control components in drone swarms, rely on inter-distances among swarm members to generate control commands. Thus, correct inter-distance information is crucial for generating safe control commands. Second, physical attacks such as sensor spoofing can cause drones to deviate from their missions, leading to changes in the inter-distance.

As a result, an attacker, who can perform physical attacks in a swarm member (target drone), can manipulate inter-distances between target drones and other swarm members, thus indirectly influencing the control commands in drone swarms. Specifically, the attacker uses *sensor spoofing* as the attack vector, and has an attack goal of high safety consequence in drone swarms: cause *another* swarm member (victim drone) that is *not* under attack, to *crash* into a front obstacle. To achieve the goal, the attacker first launches sensor spoofing in a swarm member (target drone), causing the target drone to deviate from its mission, while avoiding collisions with other swarm members (for attack stealthiness). The deviation changes the inter-distance between the target drone and other members (victim drones) , leading to incorrect control commands generated by the swarm control algorithm. Driven by these incorrect commands, the victim drones will veer off their course, potentially resulting in a crash.

In this paper, to systematically launch faulty sensor propagation attacks in drone swarms, we design a tool - SwarmSensorFuzzer, and apply it to Swarmlab [8] - a Drone Swarm Simulator. Our results show that, with a 5m deviation in a target drone under sensor spoofing, the attack success rate for another victim drone to crash into the obstacle is 76.67%. In the absence of attacks, there

CCS '22, November 7–11, 2022, Los Angeles, CA, USA

ACM ISBN 978-1-4503-9450-5/22/11.
https://doi.org/10.1145/3548606.3563535

are no crashes and the minimal distance between the victim drc and the obstacle is $1m - 5m$. Note that a 5m deviation in the tar drone is indistinguishable from natural errors as it is within t standard offset for most GPS sensors applied in commodity dro [1]. Thus, it is not easily detectable by standard safety checks swarm members, making our attacks stealthy in nature. *To the b of our knowledge, we are the first to demonstrate that sensor spoof attacks on target drones can indirectly cause significant damage drone swarms, without causing collisions of the target drone itsel*

In summary, we make the following contributions in this pa

- Demonstrate faulty sensor propagation attacks that exploit sensor spoofing to indirectly cause disruptions (e.g., crashes) in drone swarms.
- Implement the above attack on a mainstream swarm control algorithm in a drone swarm simulator.
- We find that spoofing the position of a target drone by $5m$ is sufficient to cause *other* victim drones to crash into a front obstacle. Our attack achieves 76.67% and 93.33% success rate with $5m$ and $10m$ spoofing deviation respectively.

2 MOTIVATION AND EXAMPLE

Swarm Control Algorithm. The swarm control algorithm generates control commands (e.g., velocity commands) for each swarm member and coordinates them to achieve the mission. To carry out the mission successfully, it follows three principles [7]: (1) mission-driven, to ensure the drone swarm is moving towards the destination; (2) collision-free, to ensure there are no collisions by avoiding short inter-distances among drones/obstacles; (3) cohesive formation, to ensure the formation is maintained by avoiding long inter-distances among drones. The control command for a drone consists of three sub-commands, each for a specific principle.

For example, the control commands generated according to each principle is shown in Figure 1-(a). For principle (1), each drone should have a sub-velocity moving towards the goal (i.e., blue arrows). For principle (2), with short inter-distance between drone 1 and drone 2, repulsive sub-velocities (i.e., orange arrows) are generated to avoid collisions between them. For principle (3), with long inter-distance between drone 1 and drone 5, attractive sub-velocities (i.e., green arrows) are generated to avoid drone 5 falling behind and thus to maintain the formation.

Motivating example. When a target drone is under sensor spoofing attacks (e.g., drone 4 in Figure 1-(b), it deviates from the original location, leading to the change in inter-distances between the target drone and other drones. This inter-distance change triggers the swarm control algorithm to generate new control commands based on the above principles, which could lead to collisions. For example, drone 4 in Figure 1-(b) is farther away from drone 5 due to the attack. According to principle (3), attractive sub-velocities (i.e., green arrows) are generated. Suppose that originally drone 5 plans to avoid the obstacle (i.e., purple triangle) from its left side (i.e., black arrow). With this new sub-velocity, the overall summative velocity could point towards the obstacle (i.e., red arrow), thereby leading to collisions between drone 5 and the obstacle.

To formulate the problem, we can further classify the drones involved in this attack into two categories: (1) Drones under sensor spoofing attacks, which would deviate from the mission but not directly collide with others, are called target drones. (2) Drones that are not under sensor spoofing but are indirectly influenced by the spoofing deviation, are called victim drones. In Figure 1-(b), drone 4 is the target drone and drone 5 is the victim drone.

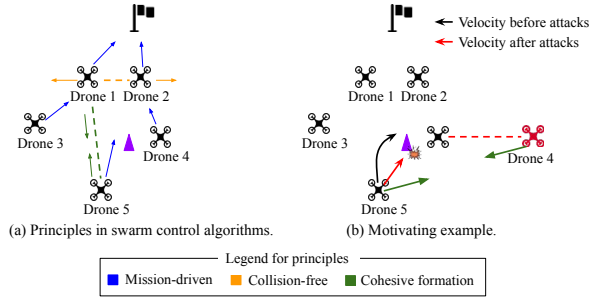

Figure 1: Principles and the motivating example behind swarm attacks.

To launch this attack, the attacker needs to choose attack parameters such as the spoofing deviation and spoofing time to cause collisions between the victim drone and the obstacle. This motivates the need for a tool to systematically discover such attacks.

3 METHODOLOGY

We design SwarmSensorFuzzer - a framework to automate faulty sensor propagation attacks. Overall, it identifies the attack parameters (i.e., the spoofing deviation and the spoofing time) by casting the attack goal as an optimization problem. Specifically, the objective function is the distance between the victim drone and the obstacle. To cause crashes, SwarmSensorFuzzer needs to find the attack parameters that minimize this objective function. As an early work, we simplify this optimization problem by performing constant deviation spoofing. To solve it, SwarmSensorFuzzer applies Gradient Descent for its effectiveness in non-linear problems. Specifically, SwarmSensorFuzzer computes the gradients of the distance with respect to the spoofing time and updates the spoofing time using these gradients. SwarmSensorFuzzer iteratively applies this process until the victim drone crashes into the obstacle.

The attacker first uses SwarmSensorFuzzer to identify the attack parameters. Then, (s)he launches the faulty sensor propagation attack in drone swarms accordingly. Specifically, the attacker performs sensor spoofing in the target drone to cause the constant deviation for certain spoofing time. This deviation changes the inter-distance between the target drone and the victim drone, triggering the swarm control algorithm to generate wrong control commands, thereby driving the victim drone to crash into the obstacle.

4 EVALUATION

4.1 Experiment Setup

Drone swarm selection. We target a highly-cited swarm control algorithm [10], which aims to navigate large flocks of drones safely in a confined space. To simulate the drone swarms and the environment, we use Swarmlab [8], a Matlab drone swarm simulator. The mission for the drone swarm is similar to Figure 1. A swarm of 5 drones aims to reach the goal while avoiding the on-path obstacle.

Attack setup. As we did not have the equipment for performing physical sensor spoofing attacks, we simulated the attacks through software code modifications - this is similar to what prior work has done [3]. Specifically, we launched GPS spoofing [9] by manipulating the GPS reading to $GPS + d$, where d is the spoofing deviation. In our evaluation, we choose d to be $5m$ and $10m$.

To perform attacks, we first use SwarmSensorFuzzer to identify the optimal attack parameters (i.e., spoofing deviation and spoofing time). Then, we launch GPS spoofing targeting a single drone in a swarm of five drones using the attack parameters.

4.2 Attack Effectiveness

Evaluation metrics. We consider the attack to be successful, if during the mission, the victim drone which is *not* under GPS spoofing crashes into a front obstacle, while there are no crashes in the absence of attacks. We do not consider the crashes caused directly by the target drone (e.g., the target drone crashes into obstacles or into other drones). We define the victim distance as the minimal distance between the victim drone and the front obstacle in the absence of attacks. This indicates how far away the victim drone is from the obstacle without attacks. We perform evaluations on 100 scenarios with each spoofing deviation setting (i.e., $5m$ and $10m$) respectively, and then report the success rate.

Results. Table 1 shows the results for two spoofing deviation settings. For missions with a victim distance lower than $5m$, the success rate is 76.67% and 93.33% for $5m$ and $10m$ spoofing respectively. This shows that our attack is quite effective, and spoofing the position of a target drone by $5m$ is sufficient to cause *other* drones to crash into a front obstacle. We observe that generally, the success rate for $10m$ spoofing is higher than that of $5m$ spoofing. This is because larger spoofing distances can trigger the swarm control algorithm to generate larger sub-velocity. Specifically, when the victim drone is approaching the obstacle, based on principle (2) in Section 2, a repulsive velocity will be generated to prevent the drone from crashing. Only if the sub-velocity triggered by the sensor spoofing is large enough to offset this repulsive velocity can the crash occur.

We also notice that for missions with a higher victim distance (i.e., $5m - 10m$), the success rate for both $5m$ and $10m$ spoofing decreases. This is because larger victim distances usually indicate larger repulsive velocities for avoiding the obstacle, which are harder to offset with the same spoofing deviation. Interestingly, we find that a $5m$ spoofing can cause crashes in missions with $5m - 10m$ victim distances. This implies that the indirect deviation in the victim drone is amplified (compared to 5m spoofing deviation). We plan to analyze reasons behind this amplification effect in the future.

Figure 2 shows the results for different values of the victim distance. With a $5m$ spoofing deviation, the success rate is above 80% when the victim distance is below $4m$. With a $10m$ spoofing distance, the success rate is above 70% when the victim distance is below $7m$. Specifically, both $5m$ and $10m$ spoofing distance can achieve 100% success rate when the victim distance is lower than $3m$. We observe that the success rate is not strictly monotonically decreasing with the victim distance, This is because factors such as the formation of the drone swarm also influence the success rate.

Table 1: Attack success rate for different spoofing deviations.

Victim Dist. (m)	Success Rate	
	$5m$ spoofing	$10m$ spoofing
$(0, 5)$	76.67%	93.33%
$(5, 10)$	15.94%	39.13%

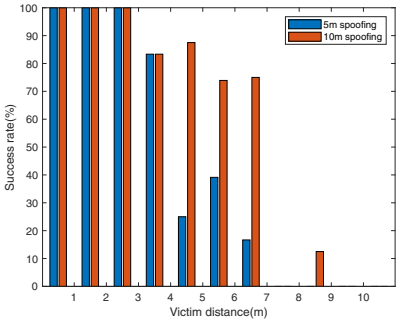

Figure 2: Attack success rate for different victim distances.

5 CONCLUSION AND FUTURE WORK

In this paper, we demonstrate faulty sensor propagation attacks that exploit sensor spoofing to indirectly cause crashes in drone swarms. We perform evaluations in a swarm control algorithm and our results show that the attack achieves 76.67% and 93.33% success rate with $5m$ and $10m$ spoofing deviation respectively. In the future, we aim to analyze the factors impacting the success rate, and extend our analysis to other swarm control algorithms.

REFERENCES

[1] 2008. Global Positioning System Standard Positioning Service Performance Standard. https://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf.
[2] 2019. *The Pentagon Wants AI-Driven Drone Swarms for Search and Rescue Ops.* https://www.nextgov.com/emerging-tech/2019/12/pentagon-wants-ai-driven-drone-swarms-search-and-rescue-ops/162113/
[3] Pritam Dash, Guanpeng Li, Zitao Chen, Mehdi Karimibiuki, and Karthik Pattabiraman. 2021. PID-Piper: Recovering Robotic Vehicles from Physical Attacks. In *51st International Conference on Dependable Systems and Networks (DSN) 2021.*
[4] Xinyu Huang, Yunzhe Tian, Yifei He, Endong Tong, Wenjia Niu, Chenyang Li, Jiqiang Liu, and Liang Chang. 2020. Exposing Spoofing Attack on Flocking-Based Unmanned Aerial Vehicle Cluster: A Threat to Swarm Intelligence. *Secur. Commun. Networks* 2020 (2020), 8889122:1–8889122:15.
[5] Chi-Gon Jung, Alipour Asl Ahad, Yuseok Jeon, and Yonghwi Kwon. 2022. SWARMFLAWFINDER: Discovering and Exploiting Logic Flaws of Swarm Algorithms. In *IEEE Symposium on Security and Privacy.*
[6] Liangjun Liu, Hongyan Qian, and Feng Hu. 2019. Random Label Based Security Authentication Mechanism for Large-Scale UAV Swarm. *2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)* (2019), 229–235.
[7] Craig W. Reynolds. 1987. Flocks, herds and schools: A distributed behavioral model. *Proceedings of the 14th annual conference on Computer graphics and interactive techniques* (1987).
[8] Enrica Soria, Fabrizio Schiano, and Dario Floreano. 2020. SwarmLab: a Matlab Drone Swarm Simulator. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2020), 8005–8011.
[9] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the requirements for successful GPS spoofing attacks. In *CCS.*
[10] Gábor Vásárhelyi, Csaba Virágh, Gergo Somorjai, Tamás Nepusz, Agoston E. Eiben, and Tamás Vicsek. 2018. Optimized flocking of autonomous drones in confined environments. *Science Robotics* 3 (2018).