False Data Injection Attacks on Unmanned Aerial Vehicles used for Package Delivery

Muhammad Iftikhar Umrani, Bernard Butler,
South East Technological University (SETU)
Waterford, Ireland

Aisling O'Driscoll,
University College Cork (UCC)
Cork, Ireland

Steven Davy
Technological University Dublin (TUD)
Dublin, Ireland

***Abstract*—This paper studies GPS and LiDAR spoofing attacks in UAV-based package delivery scenarios. An experimental testbed integrates PX4 autopilot, Gazebo simulator, and ROS to launch attacks on UAV navigation and obstacle avoidance systems. It was found that malicious software nodes can be added to generate false position and context data. Therefore, UAVs need a robust and intelligent security framework to detect attacks in real-time, to ensure successful mission completion.**

***Index Terms*—Unmanned Aerial Vehicle, Sensor Spoofing, Package Delivery, Cybersecurity, Threat Detection**

I. Introduction

Autonomous Unmanned Aerial Vehicles (UAVs), commonly known as drones, have been deployed in agriculture, search and rescue, land surveying, and notably, package delivery [1]. The integration of Artificial Intelligence (AI) and Computer Vision has greatly enhanced their operational capabilities, allowing them to operate independently rather than relying on continuous human control. The package delivery sector has attracted considerable investment. For example, Manna drone delivery [2] in Dublin, Ireland, provides timely and efficient delivery services to users, with fewer accidents.

Package delivery drones are equipped with communication modules, GPS modules, autopilot control systems, sensor modules, and onboard hardware processors. These elements enable them to gather environmental data through sensors and execute their assigned tasks. However, such UAVs have vulnerabilities in their components. UAV missions can be harmed by system faults and/or deliberate cyberattacks. System faults occur due to sensor noise or component failures. By contrast, cyberattacks are deliberate attempts to compromise the confidentiality, integrity, and availability of UAV operations. These attacks include GPS spoofing, jamming, false data injection, denial-of-service (DoS) attacks, and targeting the machine learning models used for decision-making. Both faults and cyberattacks pose significant risks to UAV operations.

The GPS signals used by UAVs lack security features like authentication and encryption, so are vulnerable to spoofing attacks. This vulnerability presents significant risk in mission-critical scenarios like autonomous package delivery, where accurate navigation and positioning are essential. Adversaries can exploit the vulnerability by injecting false data through incorrect GPS information (latitude, longitude, and altitude). Consequently, drones may misinterpret their environment and hence their position, speed and attitude, leading to crashes or collisions, particularly in fully autonomous operations. Therefore, an intelligent, secure, and resilient control system for drones in autonomous operations is needed to counter false data attacks and to ensure safe mission completion, protecting both the drones and public safety.

Current research mainly focuses on detecting security attacks, but often fails to provide effective defensive strategies or responses for detecting and mitigating multiple, coordinated attacks in real-time [3]. Given the growing complexity of UAV missions and evolution of threats, there is a clear need for solutions that can detect and neutralize such threats. Our research aims to address this gap by detecting false data injection attacks on UAVs and implementing countermeasures against such attacks. Our novel contribution lies in evaluating proposed attack strategies and their mitigation in a fully simulated, realistic UAV delivery setting. We have established a Python-based experimental testbed that includes the PX4 autopilot flight controller, the Gazebo flight simulator, and the Robot Operating System (ROS).

The scenario concerns package delivery, where a drone autonomously picks up a package, navigates around obstacles, and ultimately reaches its destination to deliver the package. We also simulated GPS and LiDAR spoofing attacks to assess their effects on the drone's navigation and obstacle detection capabilities. Experimental setup and configuration will be discussed later. This paper makes the following contributions:

1) Establish a baseline minimal configuration for normal UAV package delivery operations using an emulator.
2) Devise a way to realise selected spoofing threats.
3) Emulate GPS and LiDAR spoofing cyberattacks.
4) Identify Indicators of Compromise (like the presence of malicious ROS nodes) for these cyberattacks.

II. Related Work

Recent studies have identified vulnerabilities in UAVs related to GPS spoofing attacks and suggested better security measures. Sathaye et al. [5] developed a Real-time GPS Signal Generator (RtGSG) to generate fake GPS signals, enabling them to gain full control of the UAV and understand its behavior. Pekaric *et al* [6] simulated spoofing attacks on both GPS and LiDAR components by exploiting vulnerabilities

This research is supported by Research Ireland (formerly Science Foundation Ireland) and is co-funded under the European Regional Development Fund under Grant Number 13/RC/2077-P2 (CONNECT Phase 2)

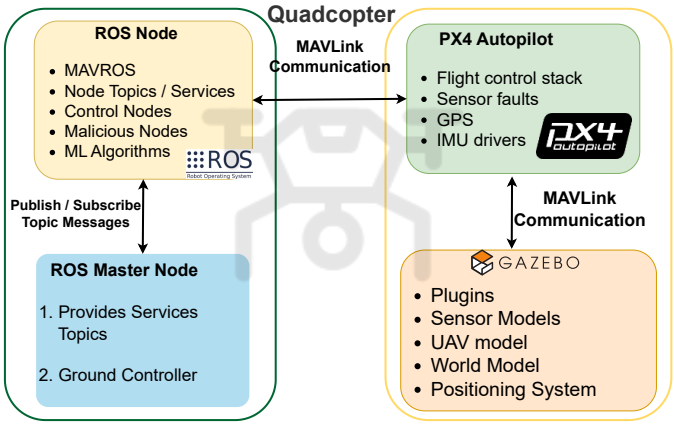

Fig. 1. Simulation Environment.

in ROS and MAVLink protocols, although their scenarios lacked real-world testing for validation. Additionally, Jung *et al.* [7] examined vulnerabilities of PX4 autopilot system and categorized different attack models. Moreover, they improved detection effectiveness through tuning the Extended Kalman Filter (EKF) gain. Seo *et al* [8] discussed the risks of using unencrypted and unauthenticated GPS signals in civilian UAVs and demonstrated how spoofing can lead to UAV navigation deviations. Moreover, they identified that an attacker needs a) to know the UAV's position and guidance system, and b) to be able to propagate signals based on the UAV's operational parameters. Lastly, Su *et al* [9] introduced a method for real-time manipulation of UAV GPS measurements, even when equipped with fault detection. Thus, emphasizing the need for refined detection techniques to identify manipulated signals. They highlight the importance of integrating simulation with real-world testing and introduced a greedy strategy for detecting falsified GPS measurements through a constrained optimization approach.

These studies have particularly focused on GPS spoofing attacks, but most have addressed only single-sensor threats. For example, while works like [5] and [6] explored GPS and, to a limited extent, LiDAR spoofing, these efforts often lacked real-time dynamic testing environments. [7] improved GPS detection through EKF tuning, but overlooked multi-sensor attacks. While [8] and [9] proposed detection methods, they were limited to GPS manipulation. Our work advances the state of the art by addressing both GPS and LiDAR spoofing in a real-time simulation environment, offering a more comprehensive and practical assessment of UAV vulnerabilities and defense strategies.

III. Simulation Environment

This section outlines the foundational elements of our emulated autonomous package delivery scenario. The various simulation components are shown in Figure 1.

A. Software in the Loop Simulation

Software-In-The-Loop (SITL) simulation replicates real-world UAV behavior, which allows testing and evaluation of UAV control algorithms, security attacks and defensive strategies without requiring physical hardware [4]. By simulating the UAV's flight dynamics, control algorithms, and sensors, researchers can assess performance, identify issues, and refine their systems in a virtual environment before their real-world implementation. This approach enhances UAV reliability and safety, minimizing cost and risks associated with testing actual UAVs.

B. Gazebo Flight Simulator

The open-source Gazebo simulator is employed to create a dynamic model of the UAV, and to build models for its sensors and cameras [10]. Gazebo offers a virtual 3D world where the physical interactions of the UAV with its surroundings can be simulated. This allows us to assess how the UAV responds to environmental factors such as wind and obstacles. Moreover, its inbuilt visualization features enable users to assess the UAV's state, including its position, orientation, and sensor readings, during its operation.

C. Robot Operating System

The Robot Operating System (ROS) is a free and open-source software framework that provides essential components, interfaces, and tools for developing advanced robotic systems [11]. Drones include actuators, sensors, and control systems, and ROS facilitates construction and seamless connection of these components, using *messages* that are exchanged in *topics* using a publish/subscribe protocol. These messages can be recorded using ROS bag files or logs. Additionally, ROS is ideal for building a UAV *digital twin*, to enhance the effectiveness and efficiency of testing and training. Furthermore, ROS supports a wide range of hardware interfaces for drone components, such as cameras, LiDAR systems, and motor controllers.

D. PX4 Autopilot

PX4 is open-source flight control software that supports many types of unmanned vehicles, including quadcopters [12]. It offers three flight modes: manual, partial-, and fully-autonomous, which allow users to choose their desired level of control. PX4 integrates essential drone components like sensors, GPS, flight controllers, and payloads, making it suitable for varied UAV operations. Furthermore, PX4 integrates seamlessly with platforms like ROS, QGroundControl, and MAVSDK, enhancing its functionality. It is also compatible with many flight simulators, allowing users to practice different drone operations in a simulated environment before actual flight.

IV. Simulation of Autonomous Package Delivery

During Gazebo-ROS simulation, *topics* receive control inputs and provide feedback on the UAV's operational state. Some topics transmit control commands and error metrics,

Fig. 2. Package delivery Scenario in Gazebo, showing a quadcopter, a multi-storey building and an unmapped obstacle (due to it being a temporary structure).

which can be used to monitor and control the flight operations. Other topics collect critical sensor data within the Gazebo simulator environment, including GPS data, IMU readings, and distance measurements, which are needed for navigation and flight stability. Each autonomy module, such as position control, attitude control, obstacle avoidance, and gripper control, is responsible for specific functions that contribute to the drone's safe and efficient operation during package delivery.

A. Baseline (minimal configuration; no attack)

In the first phase, we simulated the baseline scenario for autonomous package delivery using ROS and Gazebo, where various modules were activated by means of a launch file, to facilitate the delivery process from the source to the destination [13]. Each module is treated as a separate *ROS node*, which subscribes to and publishes messages across different topics, as listed in Table I. This communication pattern allows modules to work together in a flexible, loosely-coupled manner, coordinating actions and sharing essential data throughout the mission. The position control module manages the drone's latitude, longitude, and altitude, ensuring it reaches the desired delivery location. The attitude control module stabilizes the drone by controlling its roll, pitch, and yaw, allowing it to maintain a steady orientation throughout the flight. QR detection utilizes camera input to recognize QR codes, extracting location data that guides the drone to its drop-off point. Waypoint navigation logic directs the drone along predefined waypoints based on its position, ensuring an optimal path to the destination. Lastly, the obstacle avoidance logic employs LiDAR sensor data to detect obstacles, enabling the drone to navigate around them autonomously and to maintain a safe flight path. Figure 2 shows the drone on its way to the delivery location.

B. GPS and LiDAR Sensor Spoofing Attack

In this section, malicious attacks are added to the baseline package delivery scenario to evaluate how spoofing affects the navigation and obstacle sensing capabilities of UAVs. The impact of these attacks is illustrated in Figure 3. After successfully running the normal scenario, GPS spoofing and

TABLE I
ROS TOPICS

Topic/Service	Type	Message Type/Structure
/edrone/gps	Subscriber	NavSatFix (latitude, longitude, altitude)
/drone_command	Publisher	edrone_cmd(rcRoll, rcPitch, rcYaw, rcThrottel)
/qrValue	Subscriber	String (formatted as "latitude,longitude,altitude")
/edrone/activate_gripper	Service	Gripper (activate_gripper: bool)
/edrone/imu/data	Subscriber	Imu (orientation, angular_velocity)
/edrone/range_finder_top	Subscriber	Laserscan (ranges array with distances)
/latitude_error	Publisher	Float32 (error value)
/longitude_error	Publisher	Float32 (error value)
/altitude_error	Publisher	Float32 (error value)

LiDAR spoofing ROS nodes are added to the codebase and executed, as depicted in Table II, saving logs in a bag file for analysis.

Attack Scenario	Vulnerability	Motive	Impact
GPS Spoofing Attack	Open nature of GPS signals	Alter UAV's GPS coordinates, misguide path	Compromise navigation, risk of landing in unsafe areas, collision with obstacles
LiDAR Spoofing Attack	Vulnerable to false distance measurement signals	Manipulate LiDAR data to misrepresent surroundings	Cause UAV to misinterpret obstacles, unsafe navigation, collision, or failure in autonomous operations

Fig. 3. Threats, the vulnerabilities they exploit, their motive and their impact on package delivery

The GPS spoofing node defines original GPS coordinates as a reference and sets offsets for latitude, longitude, and altitude to create spoofed GPS data by issuing NavSatFix messages with more frequency than the genuine GPS messages. Small random variations are used to simulate realistic GPS errors so the errors are not as obvious. In parallel, the LiDAR spoofing ROS node publishes spoofed LiDAR data using fake LaserScan messages. In this case, distances are altered to simulate an environment without nearby obstacles. Such incorrect distance-to-obstacle data means that the collision avoidance system does not intervene in the flight plan, even when such intervention is needed to avoid a collision.

Both malicious ROS nodes are designed to rigorously test the robustness of the drone's navigation and obstacle avoidance systems by simulating incorrect or misleading sensor data. The objective is to identify potential vulnerabilities in the UAV's software configuration and thereby to improve their resilience against sensor spoofing attacks.

V. PRELIMINARY RESULTS

We verified the running ROS nodes and their services using the RQT graph tool in ROS. The RQT graph provides a comprehensive visual representation of all active nodes and the topics they publish or subscribe to. This enables easy tracking

TABLE II
ROS MALICIOUS TOPICS

Topic	Objective	Message Type	Spoofing Mechanism
/edrone/gps	Publish spoofed GPS data, mislead the navigation system	NavDatFix	Add random small offsets to GPS coords
/edrone/range_finder_top	Publish spoofed LiDAR data: no obstacles nearby	LaserScan	Set all range values to a large distance (e.g., 50m)

of control and communication flows within the system. During the execution of the package delivery scenario, particularly in the context of malicious attacks, we can observe the data being published by the GPS spoofing node as illustrated in Figure 4. The presence of such (unauthorised) malicious nodes in the RQT graph indicate that the system has been compromised.

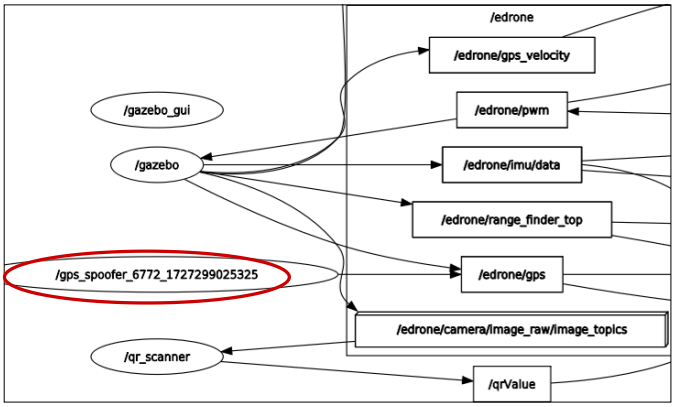

Fig. 4. RQT graph: The malicious GPS spoofing node is visible.

The LiDAR spoofing ROS node, which publishes false distance measurements, can also be seen (and hence its presence is detected) in Figure 5.

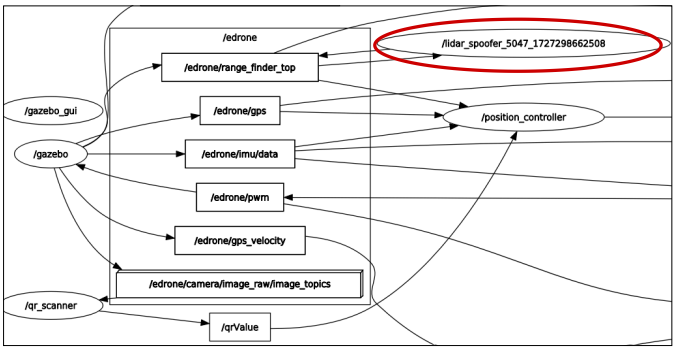

Fig. 5. RQT graph: The malicious LiDAR spoofing node is visible.

By analyzing the graphs in Figures 4 and 5, we can see how malicious ROS nodes can inject false data, but their presence can be detected from the RQT topic graph. The results show that, without safeguards like code signing and system configuration control, UAVs are vulnerable to cyberattacks that harm navigation, disrupting the mission.

Note that the assumption in this scenario is that the attacker is able to insert one or more malicious software nodes into the control software of the UAV. Hence the attacker is able to inject false data and compromise the operation of the UAV. Furthermore, the attacker is unable to hide such malicious data flows from the UAV's RQT.

VI. CONCLUSION AND FUTURE WORK

This research has demonstrated the design and implementation of an autonomous package delivery scenario using UAVs, integrating industry-standard simulation tools such as ROS, Gazebo, and PX4. The system was tested under normal conditions and subjected to malicious attacks, such as GPS and LiDAR spoofing, to evaluate its resilience and performance. Future research will focus on developing a comprehensive security framework that integrates real-time attack detection and adaptive response mechanisms. This framework will utilize anomaly detection algorithms to identify false data injection attacks in real-time and will offer dynamic reconfiguration strategies to counteract such threats. Additionally, extensive real-world testing will be conducted to validate the effectiveness of the proposed approach in operational conditions.

REFERENCES

[1] Abualigah, L., Diabat, A., Sumari, P., & Gandomi, A. H. (2021). Applications, deployments, and integration of internet of drones (iod): a review. IEEE Sensors Journal, 21(22), 25532-25546.

[2] Manna Drone Delivery, Ireland https://www.manna.aero/

[3] Mekdad, Y., Acar, A., Aris, A., El Fergougui, A., Conti, M., Lazzeretti, R., & Uluagac, S. (2024, June). Exploring Jamming and Hijacking Attacks for Micro Aerial Drones. In ICC 2024-IEEE International Conference on Communications (pp. 1939-1944). IEEE.

[4] Nguyen, K. D., & Nguyen, T. T. (2019, July). Vision-based software-in-the-loop-simulation for Unmanned Aerial Vehicles using gazebo and PX4 open source. In 2019 international conference on system science and engineering (ICSSE) (pp. 429-432). IEEE.

[5] Sathaye, H., Strohmeier, M., Lenders, V., & Ranganathan, A. (2022). An experimental study of GPS spoofing and takeover attacks on UAVs. In 31st USENIX security symposium (USENIX security 22) (pp. 3503-3520).

[6] Pekaric, I., Arnold, D., & Felderer, M. (2022, December). Simulation of sensor spoofing attacks on unmanned aerial vehicles using the gazebo simulator. In 2022 IEEE 22nd International Conference on Software Quality, Reliability, and Security Companion (QRS-C) (pp. 44-53). IEEE.

[7] Jung, J. H., Hong, M. Y., Choi, H., & Yoon, J. W. (2024). An analysis of GPS spoofing attack and efficient approach to spoofing detection in PX4. IEEE Access.

[8] Seo, S. H., Lee, B. H., Im, S. H., & Jee, G. I. (2015). Effect of spoofing on unmanned aerial vehicle using counterfeited GPS signal. Journal of Positioning, Navigation, and Timing, 4(2), 57-65.

[9] Su, J., He, J., Cheng, P., & Chen, J. (2016). A stealthy gps spoofing strategy for manipulating the trajectory of an unmanned aerial vehicle. IFAC-PapersOnLine, 49(22), 291-296.

[10] Gazebo Simulator. Retrieved from https://gazebosim.org/home

[11] O'Kane, J. M. (2014). A gentle introduction to ROS.

[12] PX4 Autopilot System. Retrieved from https://px4.io/

[13] Autonomous Package Delivery https://github.com/Project-SwaG/edrone-autonomous-ros