Tractor Beam: Safe-hijacking of Consumer Drones with Adaptive GPS Spoofing

JUHWAN NOH, YUJIN KWON, YUNMOK SON, HOCHEOL SHIN, and DOHYUN KIM, KAIST, Republic of Korea
JAEYEONG CHOI, TmaxData, Republic of Korea
YONGDAE KIM, KAIST, Republic of Korea

The consumer drone market is booming. Consumer drones are predominantly used for aerial photography; however, their use has been expanding because of their autopilot technology. Unfortunately, terrorists have also begun to use consumer drones for kamikaze bombing and reconnaissance. To protect against such threats, several companies have started "anti-drone" services that primarily focus on disrupting or incapacitating drone operations. However, the approaches employed are inadequate, because they make any drone that has intruded stop and remain over the protected area. We specify this issue by introducing the concept of *safe-hijacking*, which enables a hijacker to expel the intruding drone from the protected area remotely. As a *safe-hijacking* strategy, we investigated whether consumer drones in the autopilot mode can be hijacked via adaptive GPS spoofing. Specifically, as consumer drones activate GPS fail-safe and change their flight mode whenever a GPS error occurs, we performed black- and white-box analyses of GPS fail-safe flight mode and the following behavior after GPS signal recovery of existing consumer drones. Based on our analyses results, we developed a taxonomy of consumer drones according to these fail-safe mechanisms and designed *safe-hijacking* strategies for each drone type. Subsequently, we applied these strategies to four popular drones: DJI Phantom 3 Standard, DJI Phantom 4, Parrot Bebop 2, and 3DR Solo. The results of field experiments and software simulations verified the efficacy of our *safe-hijacking* strategies against these drones and demonstrated that the strategies can force them to move in any direction with high accuracy.

CCS Concepts: • **Security and privacy → Security in hardware**; • **Information systems → Global positioning systems**; • **Computer systems organization → Robotic autonomy**;

Additional Key Words and Phrases: Drone, anti-drone, GPS spoofing, fail-safe

ACM Reference format:
Juhwan Noh, Yujin Kwon, Yunmok Son, Hocheol Shin, Dohyun Kim, Jaeyeong Choi, and Yongdae Kim. 2019. Tractor Beam: Safe-hijacking of Consumer Drones with Adaptive GPS Spoofing. *ACM Trans. Priv. Secur.* 22, 2, Article 12 (April 2019), 26 pages.
https://doi.org/10.1145/3309735

12

The authors gratefully acknowledge the support from Nano UAV Intelligence Systems Research Laboratory at Kwangwoon University, originally funded by Defense Acquisition Program Administration (DAPA) and Agency for Defense Development (ADD).
Authors' addresses: J. Noh, Y. Kwon, Y. Son, H. Shin, D. Kim, and Y. Kim, KAIST, 291 Daehak-ro, Yuseong-gu, Daejeon, 34141, Republic of Korea; emails: {juhwan.noh, dbwls8724, yunmok00, h.c.shin, dohyunjk, yongdaek}@kaist.ac.kr; J. Choi, TmaxData, 45 Jeongjail-ro, Bundang-gu, Seongnam, Gyeonggi-do, 13613, Republic of Korea; email: jaeyeong_choi@tmax.co.kr.

2471-2566/2019/04-ART12 $15.00
https://doi.org/10.1145/3309735

1 INTRODUCTION

As drones become increasingly ubiquitous thanks to (1) cost and weight reduction of inertial measurement unit (IMU) sensors and Global Positioning System (GPS) receivers and (2) advancements in autopilot technology, they are being used for a variety of purposes such as search and rescue, natural disaster preparedness and response, and data acquisition for geographic information systems [17, 25, 26]. However, consumer drones are also being increasingly utilized by terrorists. For instance, in 2015, a Japanese anti-nuclear protester used a drone to drop a radioactive material onto the roof of Japan's prime minister's office [4]. Further, the Islamic State reportedly used drones to attack and kill troops with explosives in 2016, and DJI M600 drones carrying explosives were used to attack Venezuela's president and other officials in 2018 [53, 61].

With widespread terrorism concerns, anti-drone approaches that focus on interrupting a drone's flight have been gaining attention. Various anti-drone techniques have been proposed to protect people and property from dangerous drones. Some techniques involve shooting nets over the targeted drone to disrupt rotor operation and incapacitate drone operation [20, 36]. Other approaches use jamming to interfere with the radio control (RC) and Global Navigation Satellite System (GNSS) signals, thereby forcing the drones to hover or land [3, 22]. In addition, drones may be physically damaged with lasers to crash them [24, 60].

However, these anti-drone techniques have several disadvantages when dealing with hostile drones carrying dangerous items, such as explosives and radioactive materials. In the case of net shooters, they take too much time to capture the drone. Drone detection systems remotely detect intruding drones using radar, camera, and RF detectors. However, the range of a net shooter is only about 100m; therefore, a certain amount of time is needed to physically approach the target drone [36]. For a large airport with area in the thousands of hectares, it may take several minutes to get to the target drone, which is enough time for a dangerous drone to reach its target and explode. Laser can cause drones to crash, and radio jamming can disable remote controllers and prevent drones from moving further. However, dangerous items that pose a risk of collateral damage will still remain in the protected area that must be guarded from terrorism such as military bases and critical infrastructures. Thus, governments and militaries have become more interested in reducing the number of civilian casualties caused by the terrorism drones by removing them from the protected areas as soon as possible. In other words, advanced anti-drone technologies that can gain control of the target drone are needed. We call this advanced anti-drone technique "safe-hijacking."

Nowadays, most state-of-the-art anti-drone solutions contain a jammer as well as a drone detection system [14, 22]. To carry out a mission that evades RC jamming by anti-drone solutions, terrorists will operate drones in autopilot mode based on GPS and will not rely on a remote controller; thus, GPS spoofing is a good safe-hijacking method theoretically, because it can remotely manipulate the GPS position and velocity of drones in autopilot mode and make the autopilot navigation operate incorrectly. (It is not necessary to worry that GPS spoofing will affect aircraft even if it is used at airports, because airports halt landing and takeoff when they detect drone activity [5, 10, 45, 47, 50]. That is another reason why safe-hijacking is required in airports, because they want to expel drones and resume their services as soon as possible.) However, the existing work did not suggest a safe way to deal with dangerous and hostile drones. In 2014, Kerns et al. proposed a spoofing strategy for controlling a moving drone through GPS spoofing based on their simplistic drone model [31]. The strategy involves issuing an acceleration command to move the drone into the attacker's target path by inducing an acceleration measurement from the GPS receiver in the opposite direction through GPS spoofing signals. They performed a simulation to demonstrate their strategy and consequently showed that GPS spoofing can roughly lead a drone to move in an intended direction. However, real-world consumer drones are different from their simplistic drone model, because such consumer drones activate GPS fail-safe when a GPS error occurs. The drones

change their flight mode to the GPS fail-safe mode, and some of them subsequently recover from the fail-safe, but the simplistic drone model does not consider the GPS fail-safe mechanism. In addition, the instantaneous moving direction changes frequently; thus, it is too risky to apply their strategy to hijack dangerous drones, because it may cause the drones to crash in an environment with numerous structures or too close to aircraft.

In this study, we analyzed the GPS fail-safe mechanisms of existing consumer drones to overcome this limitation of previous studies. To this end, we performed white- and black-box analyses of four representative consumer drones, 3DR Solo [1], Parrot Bebop 2 [41], DJI Phantom 3 Standard [18], and DJI Phantom 4 [19]. The DJI drones automatically change their flight mode to the hovering mode that does not use GPS when their GPS signal reception is interrupted. They automatically change the flight mode again to another hovering mode that utilizes GPS when the GPS signal becomes available. Parrot Bebop 2 also changes its flight mode to the hovering mode like the DJI drones when it loses GPS lock, but it resumes autopilot after GPS signal recovery. 3DR Solo counts the number of occurrences of a GPS error by comparing GPS velocity and the predicted velocity through IMU sensors every 100ms using an extended Kalman filter (EKF); it activates the fail-safe when the count exceeds 10. On activation of the fail-safe, the drone changes its flight mode to hovering mode, which does not use GPS and does not recover from the fail-safe automatically. Nevertheless, the drone does not activate the fail-safe as soon as a GPS error is detected, because GPS errors may briefly occur without GPS attacks when weak or loss of GPS signals occur for a short time.

Based on the analyses results, we developed a taxonomy of consumer drones according to their fail-safe mechanisms and designed safe-hijacking strategies that allow a hijacker to gain control of each type of drones. Consumer drones can be classified into four types, and the first drone type can be hijacked by gradually moving a fake GPS position to the direction opposite to that of the desired hijacking direction, because they automatically recover from the fail-safe when they lock onto a GPS signal, even a GPS spoofing signal. The other drone types can be hijacked by manipulating their GPS location far away from their moving track. These drones are deluded as if they deviate from their track, and they will begin to move in the different direction from their original moving direction to come back to the track or follow the updated track. In particular, in the case of some of the drone types, the manipulated GPS location should be adaptively determined not to activate GPS fail-safe considering their fail-safe activation mechanism, because GPS spoofing is no longer valid to them once the fail-safe is activated. We successfully demonstrated the feasibility of these hijacking strategies on all the consumer drones that we have analyzed through field experiments and software-in-the-loop (SITL) simulations. In the experiments and the simulations, Phantom 3, Phantom 4, Parrot Bebop 2, and 3DR Solo drones moved in any desired hijacking direction with angular error magnitudes of less than 20°, 3°, 10°, and 9°, respectively. (Note that we obtained government approval for this wireless GPS spoofing and drone experiment in advance.) Demo video clips of our experiments and simulations are available at http://tractorbeam.syssec.kr.

In summary, the contributions of this study are as follows:

- We define "safe-hijacking" and analyze the fail-safe mechanism of **four** consumer drone models through white- and black-box analyses.
- Based on these analyses results, we develop a taxonomy of consumer drones according to their GPS fail-safe mechanism and design new safe-hijacking strategies for each drone type allowing a hijacker to obtain control.
- We verify the efficacy of our strategies through field experiments and SITL simulations on four real-world consumer drones. In particular, we show that our strategies can force the drones to move in any intended direction, which is beneficial in reducing the time required to remove drones from protected areas.

The remainder of this article is organized as follows. Section 2 provides background information on GPS spoofing and consumer drones. Section 3 summarizes related work. Section 4 outlines the safe-hijacking model and our assumptions. Section 5 discusses the white- and black-box analyses of GPS fail-safe mechanisms and consumer drone taxonomy according to these mechanisms. Sections 6 presents the safe-hijacking strategies for each drone type. Section 7 shows experimental and simulation results of applying these strategies to four existing consumer drones. Section 8 discusses the feasibility of safe-hijacking of other drones, mitigation of GPS spoofing for legitimate consumer drones, and legal and safety issues on GPS Spoofing. Finally, Section 9 concludes this article.

2 BACKGROUND

In this section, we provide background information about GPS systems and the two GPS spoofing approaches. The concepts of autopilot and fail-safe for consumer drones are also presented.

2.1 GPS

GPS is a satellite-based navigation system. Dozens of GPS satellites orbit the Earth and provide location and time information to GPS receivers on the ground through RF signals. As each satellite transmits its ephemeris and time of week (TOW), indicating the time at which the signal was generated, a GPS receiver can estimate its geolocation based on satellite coordinates and distances between each GPS satellite and the receiver. Each GPS satellite transmits a coarse acquisition (C/A) code signal, which civilians use, and a P(Y) code signal, which is only available for military purposes.

A civilian GPS receiver processes C/A code GPS signals in several steps. First, it determines the coarse Doppler frequency shift and code phase that represents the point where the C/A code begins. This step is called "acquisition." Next, it tracks the frequency shift and code phase as they change over time. Thus, the GPS receivers can continuously track a signal despite the multipath and Doppler effect caused by the motion of GPS satellites and the receiver. We call this state of a GPS receiver a "receiver lock" on the signal. The receiver extracts navigation data bits and measures the pseudorange, which is a distance measurement that contains atmospheric delays and clock errors, while it tracks the signals. The receiver then estimates its position, velocity, and time using the navigation data and pseudoranges. In the case of signal interruption or signal loss, the receiver will lose its lock and restart its operation from the acquisition step [6].

2.2 GPS Spoofing

Whereas P(Y) code signals are encrypted for military purposes, the C/A code signals used by civilians are neither encrypted nor authenticated. Thus, civilian GPS receivers are vulnerable to GPS spoofing. If a generated GPS spoofing signal is sufficiently strong to overshadow the authentic signal, then GPS receivers will lock onto the spoofing signal. Depending on whether they smoothly lock onto the GPS spoofing signal without interruption, the GPS spoofing is classified as soft or hard.

2.2.1 Soft GPS Spoofing. If a spoofing signal is aligned with the authentic signal that the target GPS receiver is receiving, then its operation is not interrupted, and it will lock onto the spoofing signal. We define this spoofing as soft GPS spoofing (also called seamless satellite-lock takeover) [28, 31, 44]. As illustrated in Figure 1, soft GPS spoofing involves three steps. First, a GPS spoofer generates a signal synchronized with the authentic signal that the target GPS receiver is receiving. The data carried by this spoofing signal should be same as the data carried by the authentic signal. In addition, some degree of synchronization accuracy should be ensured

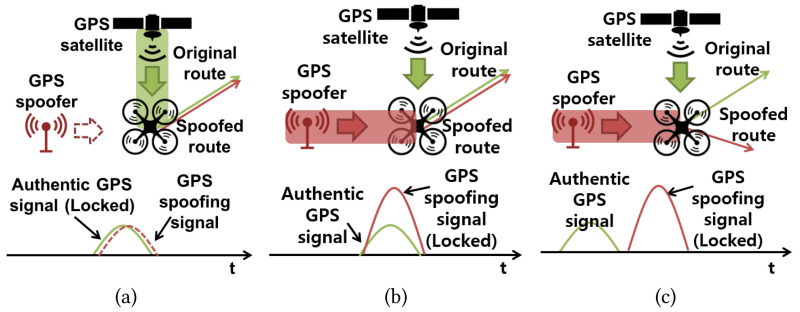


Fig. 1. Soft GPS spoofing. A spoofer (a) generates a spoofing signal aligned with the authentic signals, (b) increases the spoofing signal strength, and (c) moves the spoofed route away from the original route.

for the following parameters: (1) the power of the spoofing signal relative to the authentic signal; (2) "location offset," which is the distance between the location determined by legitimate GPS satellites and the location induced by a spoofer; and (3) "time offset," which is the delay of a spoofing signal relative to the authentic GPS signal [55]. The target receiver will lose its lock if one of these parameters does not satisfy the accuracy requirement. Next, the target receiver can smoothly lock onto the spoofing signal when the spoofer gradually increases the spoofing signal strength. Finally, the spoofer changes the spoofing signal, causing the receiver to miscalculate its location or time.

2.2.2 Hard GPS Spoofing. Soft GPS spoofing must satisfy the requirements described in Section 2.2.1. When a spoofing signal does not satisfy one of the requirements, the spoofing is called hard GPS spoofing. This spoofing signal initially acts like a jamming signal, and the target receiver may lose its lock on the authentic signal. The spoofing signal is stronger than the authentic signal. Hence, the target receiver will operate from the beginning (acquisition) and lock onto the spoofing signal at the end. However, locking onto the signal again takes tens of seconds. Thus, previous works [44, 55] claimed that loss-of-lock can be considered as the presence of GPS interference or spoofing signal. However, under normal operation, drones often encounter weak or lost GPS signals (typically in "urban canyons") [11, 33]. Thus, some consumer drones move to the fail-safe mode and wait for the lock for failure recovery instead of considering loss-of-lock as the presence of GPS interference or a spoofing signal (see Section 5 for details). Therefore, *in contrast to previous assertions, hard GPS spoofing can also be used for safe-hijacking attacks, because the failure recovery procedure, which is used to overcome the GPS shadow area, was designed without considering GPS spoofing.*

2.3 Consumer Drone System

Consumer drones run firmware that supports various advanced flight modes such as automatic takeoff and landing, hovering, and returning to launch as well as autopilot mode (i.e., Waypoints mode in the DJI drones and auto mode in ArduCopter). Consumer drone manufacturers provide a dedicated smartphone application (namely, DJI Go App for DJI drones, FreeFlight Pro for Parrot drones, and Solo App for 3DR Solo) to control the drone and manage its flight mode. For the autopilot, users can configure the drone's path before or during a flight by choosing the locations (i.e., waypoints) through the application or ground control station (GCS) software. If a user switches the flight mode to autopilot mode, then consumer drones will control their body according to their path-following algorithm in order to ensure that they travel along the path accurately despite strong winds. Users can also monitor the status of the drone such as the current flight mode and the GPS position through the application.

Consumer drones support fail-safe mode to handle emergency situations such as a crash, low battery, or loss of the remote control signal. In these situations, drones force their flight mode to fail-safe mode. For example, the fail-safe mode forces the drone to return to the start position when the battery runs low. In addition, the fail-safe mode activates an emergency stop when the drone detects a crash to reduce damage to the drone body and people near the drone.

Many consumer drones are equipped with a GNSS receiver as well as inertial measurements unit (IMU) sensors such as a gyroscope and an accelerometer. Most of these drones do not allow activating autopilot when GNSS is not available, because the accuracy and stability of the estimated position and velocity using only these sensors are not enough to operate the drones in autopilot mode, which requires precise navigation. Thus, severe GPS errors are considered an emergency situation and are handled through the fail-safe mode in consumer drones. This mechanism is designed to (1) detect situations in which the accuracy of the received GPS positions is significantly diminished or the GPS signal is continuously unstable and (2) activate a predefined action (e.g., hovering or landing).

3 RELATED WORK

Previous studies have already shown that an adversary can gain control of drones through GPS spoofing. In this section, we present several of these studies and explain why their GPS spoofing strategies are not suitable for anti-drone situations, especially defense against terrorism drones carrying explosives. We also discuss existing studies of security threats of drones that can be used with anti-drone technology.

GPS spoofing of drone systems: In 2014, Kerns et al. demonstrated that they could hijack a drone through GPS spoofing [31]. Their hijacking strategy involved making the target drone accelerate in the desired direction by inducing a fake GPS velocity in the opposite direction. They performed a simulation to demonstrate their strategy. However, the instantaneous direction of motion kept changing during the hijacking, and the direction angle was neither predictable nor controllable. Thus, this approach is too dangerous to be applied to hijacking hostile drones, because it is impossible to prevent them from crashing into nearby structures and flight vehicles. Furthermore, drones can detect GPS spoofing and activate GPS fail-safe mode by checking whether the GPS velocity is consistent with the velocity derived from the IMU sensors. Consequently, some drones may no longer rely on GPS and may not be affected by GPS spoofing due to the fail-safe mode, but Kerns et al. did not consider this possibility. They suggested a way to bypass this consistency check by adjusting the path of the attacker to make it close to the pre-planned path, but the hijacking direction changes depending on the path and is very limited. In 2015, Huang and Yang showed that a DJI drone could be forced to land by inducing GPS location of the drone to enter a no-fly zone through GPS spoofing [27]. However, this approach is not suitable for safe-hijacking requiring full control of the target drone, because this method can only force the drone to land. In 2016, Luo showed that GPS spoofing can be used to control the DJI Phantom 3 Advanced [32]. He operated the drone in "Follow Me" mode, in which the GPS location of a mobile phone is read and the drone is made to follow the provided GPS location automatically. Further, he demonstrated that he could force the drone to move without its remote controller by manipulating the GPS location of the mobile phone. In this scenario, the drone should not receive the GPS spoofing signal, because it would assume that it had arrived at the mobile phone and would not move if its GPS location was the same as the manipulated GPS location of the mobile phone due to the GPS spoofing signal. However, this method is impractical in terms of safe-hijacking, because it is very difficult to transmit a GPS spoofing signal selectively to a mobile phone without any information about the physical location of the mobile phone.

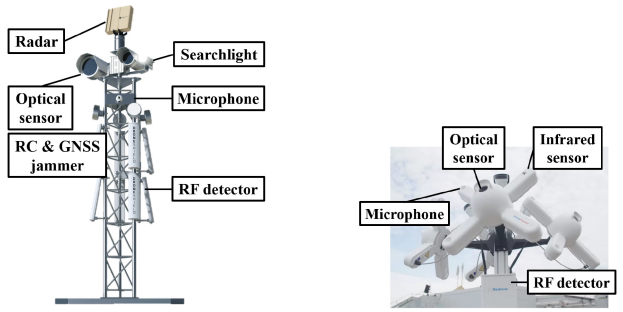


Fig. 2. Drone detection systems. (a) DroneShield's DroneSentry contains radar, optical sensor, acoustic wave sensor, and RF detector for detecting and tracking drones. It also contains an RC jammer to disable remote controllers [21]. (b) Dedrone's DroneTracker contains optical sensor, infrared sensor, acoustic wave sensor, and RF detector for detecting drones [14]. There are also plans to integrate an RC jammer to stop approaching drones [12].

Other security threats of drones: In 2012, Samland et al. empirically verified the attack vectors of AR. Drone by using unencrypted WLAN communication and open telnet and FTP [46]. Moreover, they suggested attack scenarios, such as drone hijacking, eavesdropping on video streams of AR. Drones, and tracking a specific person. However, it is not always guaranteed that every drone will have these vulnerabilities and this approach is generally applicable. In 2015, Son et al. proposed a new attack vector for drones that uses sound interference to disturb the drone's gyroscopes [48]. By generating sonic waves with the same frequency as the resonant frequency of the gyroscope in the target drone, their attack could cause a target drone to drop to the ground. However, this attack was inadequate for safe-hijacking, because it only disrupts the operation of the target drone. Based on this attack, in 2018, Chen et al. developed false data injection (FDI) attacks that manipulate EKF-based dynamic state estimation of drones by compromising a barometer or a magnetometer's output [7, 8]. Specifically, they considered EKF-based anomaly detection of drones and provided theoretical structures to bypass the detection. Even though they focused on destabilization and battery drain of drones through the FDI attacks in their studies, their approach can also be considered as complementary work that could be applied later for safe-hijacking of consumer drones as well. In 2016, Anderson developed a hardware device that can hijack drones by exploiting vulnerable remote control signals [59]. He demonstrated that an adversary can gain full control of a drone by using his device. However, the device is not applicable to drones that use other remote control protocols.

4 SAFE-HIJACKING MODEL

A hijacker requires several hardware devices and analysis of commercially available consumer drones to gain control of intruding drones through GPS spoofing. Thus, we make the following assumptions. First, our hijacking model assumes that a hijacker utilizes a drone detection system to detect and localize the intruding drones. Some anti-drone companies have already developed drone detection systems based on various types of sensors such as radar, optical sensor, and microphone. The hijacker can utilize these existing systems, such as the systems in Figure 2. Second, we assume that these systems contain an RC jammer to disable remote controllers. Terrorists can control a consumer drone manually with a remote controller during safe-hijacking; thus, we assume that an RC jammer of the drone detection system interferes with the remote control signal when the system detects a drone. To perform a mission reliably under RC jamming, terrorists will

operate drones in autopilot mode (e.g., Waypoints mode of DJI drones and the auto mode of 3DR Solo), which does not require a remote controller. In addition, they will have the advantage of reducing the risk of human error, the manpower required to operate and manipulate the drones, and the time required to complete the mission by operating drones in autopilot mode [58]. Third, we assume that the hijacker has integrated a GPS spoofer, such as a GPS simulator [37, 43, 49] and a software-defined radio (SDR)-based GPS spoofer [52], into the drone detection system, and the spoofer can adaptively generate hard and soft GPS spoofing signals according to the hijacking strategy. Fourth, we assume that the hijacker has analyzed the operational limits of the GPS chipset, fail-safe mechanism, and path-following algorithm of all commercially available consumer drones in advance of intrusion of drones, and the hijacker will build strategies for each consumer drone based on the analysis. Because the drone detection systems enable a hijacker to identify the manufacturer and model of the intruding model [13, 23], the hijacker can apply the corresponding strategy for the intruding drone according to its manufacturer and model. Note that we do not consider hijacking through the exploitation of vulnerabilities in the drone's firmware and the remote control communication protocol.

5 GPS FAIL-SAFE MECHANISMS OF CONSUMER DRONES

To design general safe-hijacking strategies applicable to most consumer drones based on GPS spoofing, it is necessary to understand the GPS fail-safe mechanisms of consumer drones, because severe GPS errors caused by GPS spoofing can activate the GPS fail-safe mode. This section presents our analysis results for the fail-safe mechanisms of DJI Phantom 3 Standard, DJI Phantom 4, Parrot Bebop 2, and 3DR Solo, which are representative consumer drones. Dynamic analyses were performed to understand the GPS fail-safe mechanisms of these drones by transmitting hard GPS spoofing signal to them. In addition, we investigated the publicly available source code of 3DR Solo, and extensively analyzed the EKF failure detection algorithm. Furthermore, we decomposed consumer drones into four types according to their fail-safe mechanisms based on our analysis results. This classification will facilitate the design of general safe-hijacking strategies that can be applied to other consumer drones not analyzed in this study.

5.1 DJI Phantom 3 Standard, Phantom 4, and Parrot Bebop 2

The user manuals of the DJI Phantom 3 Standard, Phantom 4, and Parrot Bebop 2 describe the RC fail-safe to deal with interruption of the remote control signal, but do not state whether the drones support GPS fail-safe [15, 16, 40]. Therefore, we analyzed the GPS fail-safe of the drones by observing their respective flight mode and GPS status through DJI Go app and Parrot FreeFlight Pro app, a mobile application that pilots the drones, when their GPS signal reception is interrupted by a hard GPS spoofing signal. Our experimental setup is presented in Figure 3.

DJI Phantom 3 Standard lost its lock on the existing GPS satellite signal when we transmitted a hard GPS spoofing signal to it in Waypoints mode. Its GPS fail-safe behavior is to change its flight mode to ATTI mode, which uses only its barometer to maintain its altitude and does not depend on GPS. After a while, the GPS position of the drone was manipulated. It then automatically recovered its mode to F-GPS mode, which is the positioning mode that uses both GPS and its barometer to stably hover. In other words, DJI Phantom 3 Standard supports GPS fail-safe that enables it to hover and not rely on GPS. Further, it automatically recovers from the fail-safe when it locks onto any GPS signal even a GPS spoofing signal.

As with DJI Phantom 3 Standard, DJI Phantom 4 lost its lock on the GPS signals and automatically changed its mode to OPTI mode, which uses its optical sensors, when we transmitted a hard GPS spoofing signal to the drone operating in Waypoints mode. *However, DJI Phantom 4 did not recover from the fail-safe even though we kept the hard GPS spoofer on.* We surmised why it did not

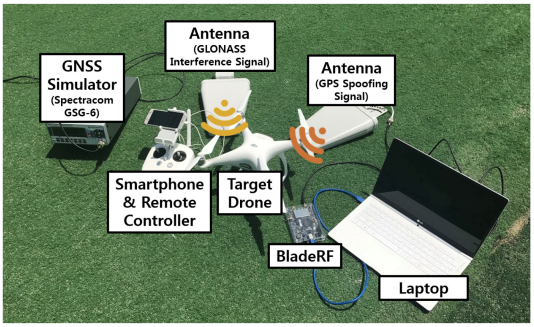


Fig. 3. Experimental setup consisting of a laptop, an SDR device, and a directional antenna. The laptop can generate GPS baseband interference and spoofing signals, and BladeRF converts their frequency into the L1 band (i.e., 1575.42MHz). This spoofing signal was transmitted to DJI Phantom 3, Phantom 4, Parrot Bebop 2, and 3DR Solo. In addition, a GLONASS interference signal was generated for the analysis and experiments on DJI Phantom 4 by using a GNSS simulator. The experiment was conducted with government approval.

(a) (b) (c)

Fig. 4. Analysis of DJI Phantom 4: (a) the drone was receiving authentic GNSS signals (GPS and GLONASS). (b) We generated a hard GPS spoofing signal. According to the GPS status icon in the DJI Go app, the drone was still receiving a weak signal from four satellites. (c) We generated GLONASS interference signal. After a brief loss-of-lock, the drone locked onto the GPS spoofing signal.

recover through the GPS status icon on the DJI Go app, showing the satellite signal strength and the number of satellites that the drone is locked onto, because it was the only information that we could obtain about its GNSS receiver status. The status showed that it was still receiving weak signals from some satellites even though we transmitted a hard GPS spoofing signal (Figure 4(b)). We hypothesized that the hard GPS spoofing attack failed, because the drone used GLONASS and GPS [16]. As mentioned in Section 2.2.2, the GPS receiver locks onto a hard GPS spoofing signal as it operates from the beginning. However, the receiver still seemed to be tracking the signal from some satellites, assumed to be GLONASS satellites. We examined our hypothesis by using a Spectracom GSG-6 GNSS simulator that supports both GPS and GLONASS signal simulations. During the hard GPS spoofing attack, we generated a GLONASS interference signal by using the GNSS simulator and transmitted it to Phantom 4. Phantom 4 successfully locked onto the hard GPS spoofing signal after a brief loss-of-lock (Figure 4(c)), then it changed its mode to the GPS mode that utilizes GPS and GLONASS to stably hover. Although we utilized an expensive and bulky GNSS simulator to generate the GLONASS interference signal for these analyses and experiments, a hijacker can utilize inexpensive and portable GLONASS jammers available in the online marketplace [30].

We also transmitted a hard GPS spoofing signal to Parrot Bebop 2 while it was operating in Flight Plan mode, a similar flight mode to Waypoints mode in the DJI drones. As a result, it lost its lock on GPS, and it began to stably hover. We continued signal transmission and observed that the drone locked onto the spoofing signal after a minute. In contrast to the DJI drones, it resumed its Flight Plan mode, and it kept performing its mission and tracking its pre-defined trajectory.

5.2 3DR Solo

The 3DR Solo software is based on ArduCopter [2], the source code and documentation of which are publicly available. We performed a dynamic analysis of the 3DR Solo and analyzed its source code to understand how it deals with GPS spoofing and the fail-safe mechanism during autopilot.

5.2.1 Dynamic Analysis of the Fail-safe Mechanism. ArduCopter supports the autopilot in "auto mode." Users can make drones takeoff, land, and move to a waypoint, and a drone automatically passes its waypoints and reaches its destination. Users can send missions to the drone via the app. In auto mode, the drone follows the mission based on the position estimated by the EKF algorithm.

We first performed dynamic analysis of 3DR Solo in auto mode and transmitted a hard GPS spoofing signal to understand its behavior to a hard GPS spoofing attack. We observed the flight mode and GPS status of the drone through QGroundControl, the mobile application for ArduCopter. We determined that the drone lost its lock on the existing GPS satellite signal and changed its mode to Alt Hold mode, maintaining its altitude by only using a barometer when we transmitted the signal. In addition, QGroundControl triggered a voice alarm and a log message showing "EKF variance error." After a while, the drone locked onto the spoofing signal, but Alt Hold mode does not utilize a GPS receiver. Thus, hard GPS spoofing signals cannot affect the drone until the pilot manually changes its mode to a GPS dependent mode.

5.2.2 Static Analysis of the EKF Failure Detection. 3DR Solo contains sensors (i.e., gyroscope, accelerometer, and GPS receiver). The EKF algorithm of ArduCopter estimates various parameters, including velocity, position, and magnetic field by fusing the IMUs' output with GPS measurements. The EKF algorithm starts by predicting its position and velocity by only using the IMUs' outputs. The IMUs contain inherent errors. Hence, the EKF periodically measures its position and velocity innovations, which are the differences between the prediction and GPS measurements, in the north-east-down coordinate system. The EKF algorithm then applies these innovations to correct the drone's position and velocity. During this process, ArduCopter checks if the variance of the EKF exceeds a pre-defined threshold value. Algorithm 1 describes the EKF failure detection mechanism at a high level.

The task `ekf_check()` is called by the ArduCopter scheduler every 100*ms*. If the drone's power is on but its motors are stopped (line 2), then it will initialize (1) a flag, bad_variance, indicating whether an EKF fail-safe event has occurred and (2) a counter, fail_count, that will be increased when it detects the EKF failure (lines 3 and 4). If the motors are working, then this task will obtain two values from the EKF. The first value is innovVelSumSq, the sum of the square of the velocity innovation in each axis (line 6). The second is varVelSum, the sum of the velocity innovation variance in each axis (line 7). The variance of the EKF is defined as line 8 based on these values. If the variance exceeds 0.8 (line 9) and the bad_variance is not yet flagged (line 10), then it will be considered as an EKF failure. If fail_count exceeds 10 (line 12), then the bad_variance will be flagged as True (line 14) and the EKF fail-safe mode—Alt Hold mode by default—will be activated (line 15). In summary, the EKF fail-safe mode will be activated if the drone GPS receiver loses its lock or if an inconsistency is found between the velocity estimated by the IMUs and the GPS velocity for at least 1 *s*.

ALGORITHM 1: EKF failure detection algorithm

```
// ekf_check() is called at 10  Hz by ArduCopter scheduler
Function ekf_check()
    if motors are stopped then
        bad_variance ←False
        fail_count ← 0
        return
    get innovVelSumSq from the EKF
    get varVelSum from the EKF
    velVar ←sqrt(innovVelSumSq / varVelSum)
    if velVar >= 0.8 then
        if bad_variance is not True then
            fail_count ←fail_count +1
            if fail_count >= 10 then
                fail_count ← 10
                bad_variance ←True
                change its mode to the EKF fail-safe mode
    else
        if fail_count > 0 then
            fail_count ←fail_count −1
            if bad_variance is True and fail_count == 0 then
                bad_variance ←False
```

5.3 Taxonomy of Consumer Drones

When designing GPS spoofing-based safe-hijacking strategies, the GPS fail-safe mechanism of the target drone should be considered. Thus, to generalize safe-hijacking strategies that cover consumer drones not mentioned in this article, we need a taxonomy of consumer drones according to GPS fail-safe mechanisms, and the corresponding safe-hijacking strategy for each fail-safe type of drone should be developed.

Basically, MEMS sensors, mainly used in consumer drones as IMUs, are subject to sources or error such as misalignment and temperature change. In reality, most consumer drones do not allow to activate autopilot without GPS. Thus, GPS fail-safe flight mode of consumer drones should be either the positioning mode, which does not utilize GPS, or landing mode, because it is too dangerous to maintain autopilot based on only IMUs without GPS. In the case of drones whose GPS fail-safe mode is the positioning mode, there exists three different ways after the drones lock onto GPS signal again, and we can classify consumer drones according to these behavior types. The first type is to switch from the fail-safe mode to positioning mode, which utilizes GPS, as done by the DJI drones. The second type is to resume their autopilot just before loss-of-lock, as done by Parrot Bebop 2. The third type is to maintain GPS fail-safe even though GPS is available and wait until a pilot gives a new command as done by 3DR Solo. Including the case of drones whose GPS fail-safe mode is landing mode, our classification scheme is summarized in Table 1.

6 SAFE-HIJACKING STRATEGY

Section 5 analyzed the GPS fail-safe mechanisms of four representative consumer drones and classified consumer drones into four types according to their fail-safe mechanism based on the

Table 1. Classification of Consumer Drones According to Their GPS Fail-safe Mechanisms and the Corresponding Safe-hijacking Strategy for Each Consumer Drone Type

Drone type	GPS fail-safe flight mode	Behavior after GPS recovery	Corresponding safe-hijacking strategy	Belonging consumer drones
I	Positioning mode (non-GPS)	Positioning mode (GPS)	Strategy A	DJI Phantom 3 & Phantom 4
II		Autopilot (GPS)	Strategy B	Parrot Bebop 2
III		Continue fail-safe	Strategy C	3DR Solo
IV	Landing			—*

* We were not able to find any consumer drones that correspond to Type IV. We added Type IV for the sake of completeness without any missing consumer drone.

analysis. This section introduces our safe-hijacking strategies for each drone type. We deal with the general outline of each strategy in this section, and Section 7 discusses how these strategies can be applied to the existing consumer drones in detail.

6.1 Strategy A: Against Type I Drones

As mentioned in Section 5.3, we define Type I as drones that try to stay in place using a GPS signal, also called GPS positioning mode, even if the drone recovers from the GPS fail-safe flight mode. The main difference between Type I and Types II, III, and IV is that Type I also uses GPS to stably hover.

Strategy A deals with Type I drones. The key to this strategy is that the Type I drones are trying to stay over their original position. If the attacker spoofs the target drone's GPS position as if the drone is moving in a certain direction, then drones are considered to drift owing to external factors such as wind. Thus, the target drone generates speed in the opposite direction, so the drone moves in that direction in the real world. For example, if the attacker spoofs the target drone's GPS position to eastward gradually, the target drone will move to the west. With this method, GPS spoofing is possible in all directions—360 degrees for Type I drones. We will discuss how to apply Strategy A to Type I consumer drones, DJI Phantom 3 and 4 in this work, and its safe-hijacking accuracy in Section 7.1.

6.2 Strategy B: Against Type II Drones

Type II drones also activate the fail-safe mode when they lose GPS lock, but they resume their autopilot, unlike Type I drones, when they lock onto GPS again. Strategy A cannot be applied to hijack these drones safely, because they will accelerate to the hijacking direction while they try to move to the next waypoint. Therefore, we need another strategy to hijack Type II drones safely.

Our new strategy is based on the drone's characteristic that control their body according to their path-following algorithm, which enables them to reach their destination reliably and accurately during autopilot. If the GPS position is manipulated as the drone deviates from the path, then it will move in a different direction from the original direction to return to the track. The moving direction is determined by their path following algorithm and the fake position; therefore, the algorithm of Type II drones should be analyzed in advance before safe-hijacking. Then, based on the analysis of the path-following algorithm, the hijacker can determine the hijacking direction and calculate the corresponding fake location. We will discuss how to determine the fake location and safely hijack Parrot Bebop 2, a Type II drone, and its safe-hijacking accuracy in Section 7.2.

6.3 Strategy C: Against Type III and IV Drones

As discussed in Section 5.3 and presented in Table 1, Type III and IV drones are characterized by no automatic recovery from fail-safe. This means that once a Type III or IV drone loses GPS lock and

gets into the fail-safe mode, there is no way to affect it by any manipulation of fake GPS signals. Therefore, for these types of drones, it is of utmost importance in the hijack strategy not to make them fall into the fail-safe mode, where a special hijack strategy different from Strategies A and B is required.

Strategy C deals with this special type of drones. In this strategy, the focus is to prevent the target drone from losing GPS lock despite GPS signal manipulations. Although the specific extent may vary according to the GPS receiver used, the spoofed GPS signals must be consistent with the original ones that the target drone have been receiving before hijacking. This necessitates some degree of soft GPS spoofing, where the GPS spoofer first generates fake signals synchronized to the authentic signals and gradually deviates the spoofed location from the real. Because there is no interruption or loss of lock during this procedure, the target drone remains GPS-dependent even after it fully locks onto the spoofed signal. Once the target drone locks onto the spoofed signal, the next step is to move the spoofed location accordingly. The way the spoofed location should be moved to control the target can depend on various factors: path-following algorithm used, adoption of secondary GPS-independent source of location (e.g., IMU or inertial navigation system), how sensitive the GPS receiver is, and so on. Therefore, there can be numerous variations of Strategy C—possibly as many as the number of drone models. For example, 3DR Solo, analyzed in Section 5.2, utilizes an EKF algorithm that fuses GPS and IMU measurements together to estimate its location, and which also continuously monitors the consistency between the two with a tolerance—fail_count—of 10. Therefore, Strategy C adaptation to 3DR Solo must carefully handle the spoofed location such that it does not deviate from the real location—a safe alternative to the location estimation by IMU, because the hijacker cannot know it—consecutively 10 times. We will revisit the 3DR Solo example in Section 7.3.

In summary, Strategy C is a safe-hijacking technique applicable to drones that never recover from GPS-independent fail-safe mode. It is composed of two steps: (1) conducting soft GPS spoofing to make the target seamlessly lock onto the spoofed signal and (2) moving the spoofed location adequately according to the characteristics of the target drone is characterized. Note that Strategy C can also be applied to Type I and II drones. However, it is the decision of the hijacker whether to apply this strategy to those types of drones, because soft GPS spoofing is more difficult and expensive than hard GPS spoofing. The hijacker may apply Strategy A or B if the target can be fully identified as Type I or II, and apply Strategy C when the target is of Type III or Type IV or is not identifiable.

7 EXPERIMENTS

This section shows how the strategies described in Section 6 can be applied to existing consumer drones and presents experimental results that measure hijacking accuracy.

7.1 Case Study for Strategy A: Against DJI Phantom 3 and 4

7.1.1 Safe-hijacking Strategy. A hijacker can cause the DJI drones to rely on a hard GPS spoofing signal based on our black-box analysis of GPS fail-safe in Section 5.1. This means that the GPS position and velocity of the drones can be manipulated as intended by the hard GPS spoofing signal. We tried to manipulate GPS velocity while the drones were in F-GPS mode (Phantom 3) or GPS mode (Phantom 4), and confirmed that they moved in the opposite direction when we consistently induced the fabricated GPS velocity in a specific direction as discussed in Section 6.1.

We designed a safe-hijacking strategy based on the abovementioned characteristic of the drones and our black-box analysis. First, we stopped Waypoints mode operation of the drone by transmitting hard GPS spoofing signals. We also transmitted GLONASS interference signals in the case of Phantom 4. Second, we continuously transmitted the spoofing signals with strength greater than

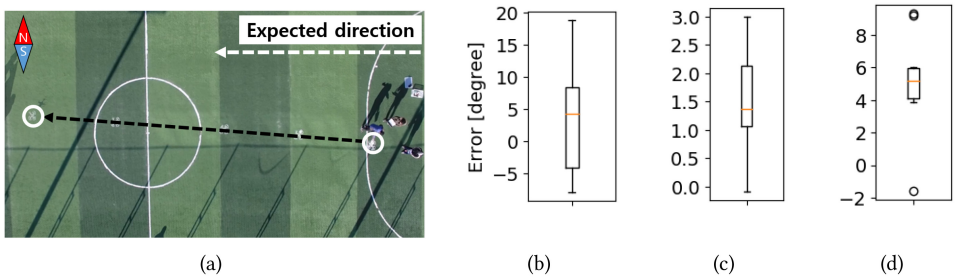


Fig. 5. Results of the safe-hijacking experiment on DJI drones and Parrot Bebop 2: (a) We shot bird's-eye view videos during the experiment by using a high-flying drone and measured their moving direction from the video clips. (b) Box plot of the angular errors from the expected direction (DJI Phantom 3 Standard). (c) Box plot of the angular errors from the expected direction (DJI Phantom 4). (d) Box plot of the angular errors from the expected direction (Parrot Bebop 2).

that of the authentic GPS signals to make the drone lock onto the spoofing signals. Third, we induced its GPS velocity in the direction opposite to that in which we wanted to make it move when the target drone locked onto the spoofing signals.

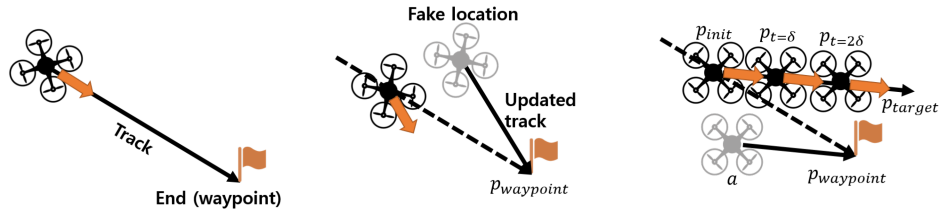
7.1.2 Experimental Results. We repeated this safe-hijacking experiment 10 times on each drone to demonstrate its feasibility according to our strategy. Figure 5(a) explains how we measured the moving direction of the drones, and Figure 5(b) and (c) show the experimental results obtained. We induced Phantom 3 to calculate its velocity after the drone locked onto the spoofing signals, indicating that it was moving eastward and expecting it to move westward. We induced the velocity westward with the expectation that the drone would move eastward in the case of Phantom 4.

Theoretically, the drones have to move straight westward or eastward through safe-hijacking. However, in practice, an angular error exists between the expected direction and the measured path, which may be caused by the internal noise (e.g., IMU errors) of the drone or external environmental factors, such as wind. However, the angular errors are reasonable, and the drone's moving path was straight during safe-hijacking. Hence, the hijacker can approximately control the target drone as he/she wants. Figure 5(b) presents a box plot of the angular errors of Phantom 3. The angular errors were at most approximately 19°. All results were within ±10° except for those of two experiments. In the case of Phantom 4, the angular error values were less than 3°. In other words, a hijacker can more accurately drive Phantom 4 than Phantom 3. The extra sensors, such as the camera and ultrasonic sensors in Phantom 4, seemed to improve its flight stability. These angular errors may increase the position error between the desired and actual locations as the travel distance increases. However, the hijacker can reduce these errors by adjusting the spoofing direction during safe-hijacking.

We also tried to apply the safe-hijacking strategy of Phantom 4 to DJI Mavic Pro, which, like Phantom 4, utilizes both GPS and GLONASS. Consequently, we observed that it also moved in the intended direction similar to Phantom 4. This demo video clip is also available at http://tractorbeam.syssec.kr. This safe-hijacking strategy is expected to be applicable to other DJI drones that utilize both GPS and GLONASS.

7.2 Case Study for Strategy B: Against Parrot Bebop 2

7.2.1 Path-following Algorithm and Safe-hijacking Strategy. Parrot Bebop 2 also provides flight plan mode, which is similar to the Waypoints mode of DJI drones. To determine the path-following algorithm of Parrot Bebop 2, we manipulated the GPS location of the drone out of its track via hard



(a) (b) (c)

Fig. 6. Path-following algorithm of Parrot Bebop 2 in flight plan mode. (a) It determines its moving track from its location to the next waypoint when it starts flight plan mode. (b) It continuously updates its track from its current location to the next waypoint, and the location of the next waypoint $p_{waypoint}$ is the intersection point of the original track and the updated track. (c) The direction from the manipulated location to $p_{waypoint}$ becomes the hijacking direction.

GPS spoofing while it operated in flight plan mode and observed its moving direction. As a result, we found out that its moving direction is always the direction from the fake location to the next waypoint. This means that the drone keeps updating its track from its current location to the next waypoint while it is operating in flight plan mode.

By using this path-following mechanism, the drone can safely be hijacked in any intended direction. The safe-hijacking strategy is as follows. First, the hijacker finds the location of the waypoint $p_{waypoint}$ by manipulating the GPS location out of the track as described in Figure 6(b). The moving direction of the drone is the direction of the updated track according to the path-following, so the updated track is the straight line from the fake location to this direction, and the waypoint will be located at the intersection point of the original track and the updated track. Second, the hijacker determines the fake location and generates a GPS spoofing signal that manipulates the GPS location to the fake location. We assumed that the hijacker wants the drone to move in the direction toward the target position p_{target}. The fake location should be a point on the line passing through the waypoint in the intended hijacking direction as described in the following equation and Figure 6(c),

$$a = p_{waypoint} + k \cdot (p_{target} - p_{init}), \quad \text{where } k < 0.$$

The equation is the vector form of the parametric equation of a line, and k is a free parameter that can be any negative number.

7.2.2 Experimental Results. We operated the drone in flight plan mode and set its destination to the penalty spot of the mini football field. We generated a hard GPS spoofing signal that manipulates the GPS location to 20m due south of the penalty spot, such that the expected moving direction was northward. In the same experimental setup as in Section 7.1, we applied the safe-hijacking strategy and repeated it 10 times.

We observed that the drone always moved north. Figure 5(d) summarizes the experimental results. The angular errors were at most 10°, and the average angular error is 5.13°. In this experiment, we assumed the hijacker already knew the location of the destination, but in practice, there will be estimation errors that will adversely affect the hijacking direction accuracy.

7.3 Case Study for Strategy C: Against 3DR Solo

From our analysis of 3DR Solo in Section 5.2, we discovered that it never recovers from fail-safe mode once the preset EKF failure count of 10 is exceeded. This indicates that 3DR Solo is a Type III drone, which all require Strategy C for safe-hijacking. Although, as discussed in Section 6.3, Strategy C can have numerous variations according to how the target drone is implemented, the

following characteristics of 3DR Solo makes it a comprehensive example of drones to which Strategy C has to be applied:

- It has a complicated path-following algorithm that is characterized by Intermediate Target Position (ITP), which necessitates moving the spoofed GPS location accordingly to safely control the behavior of the target drone.
- It utilizes the EKF fail-safe mechanism, which fuses the GPS and IMU output together. This practically makes its IMU a secondary source of location, and a hijacker is only allowed to perturb the GPS location with the margin specified by the mechanism.
- Its GPS receiver has an upper limit on how much the GPS location can be updated at once without losing the lock. This limit acts as a hard restriction on how the hijacker has to move the spoofed location.

Therefore, we chose 3DR Solo for the case study of Strategy C. We analyzed the source code of ArduCopter, on which 3DR Solo is based, to understand its path-following algorithm. Subsequently, based on this analysis, we built a safe-hijacking strategy that handles all three characteristics presented above.

7.3.1 Path-Following Algorithm. The drone tries to move to the next waypoint after taking off in auto mode. During its flight, the drone can deviate from the track because of external influences such as wind disturbance. Thus, ArduCopter handles this situation to prevent mission failure based on its path-following algorithm. We analyzed the source code of the path-following of ArduCopter to understand how it behaves when it deviates from its track.

ALGORITHM 2: ArduCopter path-following algorithm

```
// The original name of the function is advance_wp_target_along_track()
// advance_target() is called at 400 Hz by the ArduCopter scheduler
Function advance_target()
    // ITP is initialized with the starting point of the track
    get track from the mission uploaded by a user
    location ←the coordinates of the current location from the EKF
    distance ←the distance from the location to the track
    if distance is less than 13 m then /* Zone 0 */
        ITP advances slightly forward along the track;
    else
        perpendicular_foot ←the coordinates of perpendicular foot from the location to the track
        if ITP is closer to the end of the track than perpendicular_foot then /* Zone 1 */
            ITP stays
        else
            if distance / (the distance between location and ITP ) > 0.98 then /* Zone 2 */
                ITP advances slightly forward along the track
            else /* Zone 3 */
                ITP advances slightly forward along the track until the drone's speed along the track
                exceeds 1 m/s
                ITP halts when the drone's speed along the track exceeds 1 m/s
    controls motors to move the drone body to ITP
```

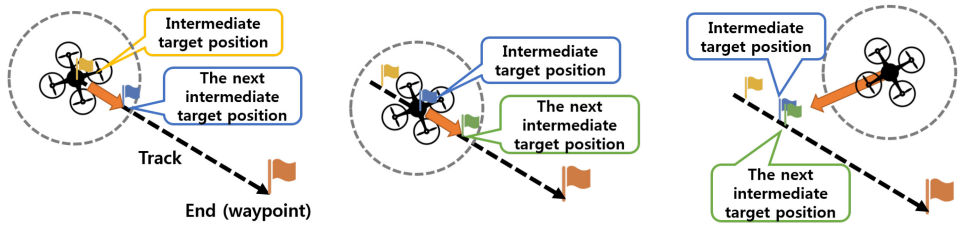


Fig. 7. Path-following algorithm in auto mode. (a) It checks that the track is reachable within a specific distance, which is called the leash length. (b) The ITP is updated to the next ITP if the leash reaches the track. (c) If the track is not reachable, then the ITP will stay or advance slightly.

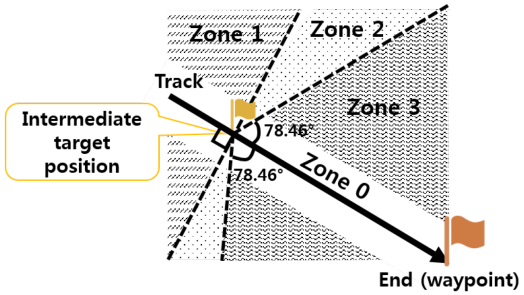


Fig. 8. Area classification for analysis of the position control algorithm. From a drone, zone 0 is the area in which the track is reachable within the leash length of 13 m. Zone 1 is the area in which the drone's perpendicular foot is farther to the end of the track than the ITP. Zone 2 is the area in which the perpendicular foot is not farther, but the ratio of the distance to the distance between the location and the ITP exceeds 0.98. Zone 3 satisfies the same condition as Zone 2, but the ratio is less than 0.98.

ArduCopter builds its track from its origin to the next waypoint. Specifically, ArduCopter periodically advances the Intermediate Target Position (ITP) along the track in small increments and causes the drone body to move to the ITP (Figure 7(a) and (b)). Thus, although the drone deviates from the track, it will return to the track, because the ITP is always on the track. If the distance between the drone and its track exceeds a specific value (specifically, 13m), which is called the leash length, then the ITP will be updated to a slightly advanced position from the last position to the end of the track (Figure 7(c)). Otherwise, the ITP may be retained or moved, depending on where the drone is located.

Algorithm 2 describes how ArduCopter advances the ITP based on its location. The position control algorithm obtains the current track from the mission (line 2), and its location from its EKF (line 3). The system measures the distance from the location to the track (line 4). The ITP will advance slightly along the track (line 6) if distance is less than 13 m (line 5); otherwise, it will estimate the perpendicular_foot from the location to the track (line 8). The ITP will be maintained (line 10) if it is closer to the end of the track than to the perpendicular_foot (line 9); otherwise, the algorithm will consider if the ratio of the distance to the distance between the location and the ITP exceeds 0.98 (line 12). If it does, then the ITP will advance, but soon halt (line 13), because the ITP will become closer to the end of the track than the perpendicular_foot (line 9); otherwise, the ITP will halt if the drone's speed along the track exceeds $1m/s$ (line 15). Finally, the drone controls its motors to reach this updated ITP. Figure 8 illustrates the four cases incorporated in Algorithm 2.

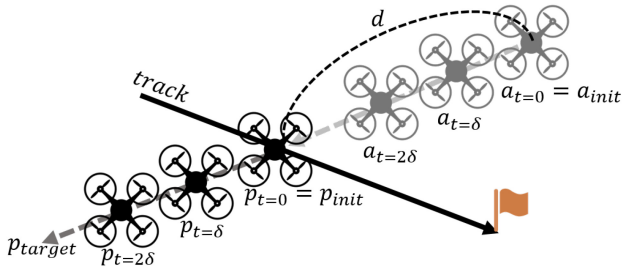


Fig. 9. At time 0, safe-hijacking is initiated by spoofing the GPS location of the target drone to a_{init}. During the short time δ, the moving direction and the distance of $a_{t=\delta}$ should be equal to those of $p_{t=\delta}$ to avoid activating the EKF fail-safe.

In summary, the ITP will remain unchanged if the drone is in Zone 1 or advance slightly forward along the track for a while and eventually halt if the drone is in Zone 2 or 3. Thus, from a GPS spoofing attacker's point of view, the ITP can be estimated to be the physical location of the drone immediately prior to GPS spoofing, and the movement of the drone is predictable based on this analysis.

7.3.2 Safe-hijacking Strategy. If the drone mistakenly determines that it has deviated from the track owing to GPS spoofing, then it will move in the direction away from the manipulated location to the ITP, as discussed in Section 7.3.1. Thus, the drone can be hijacked in an intended direction by manipulating its GPS location considering the ITP and the hijacking direction.

(i) *Initial Fake Location.* The ITP is fixed or only slightly changed when the GPS location deviates from the track by more than the leash length. Hence, we can say that the physical location of the target drone immediately before safe-hijacking is an approximation of the updated ITP. Thus, we can derive the possible coordinates of the initial fake location from the approximated ITP and the intended hijacking direction.

We assume that the hijacker begins to hijack the target drone at the physical location $p_{init} = (p_1, p_2, p_3)$ and wants the drone to move in the direction toward the target position p_{target} (Figure 9). Thus, we can define the hijacking direction, which is the direction of the line from p_{init} to p_{target}. The Earth-centered Earth-fixed coordinates of the location $p_{init} = (p_1, p_2, p_3)$ can be derived from the latitude and longitude of p_{init}, which are given as α and β in degrees [9]. (The terms e and a are the eccentricity and Earth's radius at the equator, respectively. The altitude of the drone is negligible compared to the radius of the Earth.)

$$
\begin{aligned}
p_1 &= R \cdot \cos(\alpha\pi/180) \cdot \cos(\beta\pi/180) \\
p_2 &= R \cdot \cos(\alpha\pi/180) \cdot \sin(\beta\pi/180) \\
p_3 &= (1 - e^2)R \cdot \sin(\alpha\pi/180), \quad \text{where } R = a/\sqrt{1 - e^2 \sin^2 \alpha}
\end{aligned}
$$

According to its path-following algorithm, the drone will try to move to the ITP (approximately equal to p_{init}) if its GPS location deviates from the track by more than the leash length. Thus, the direction from the initial fake location $a_{init} = (a_1, a_2, a_3)$ to the ITP (approximately equal to p_{init}) should be the same as the hijacking direction. In addition, the distance between a_{init} and p_{init} should exceed the leash length (i.e., 13m) to fix the ITP. We can formulate these requirements as the following equation and, consequently, can hijack 3DR Solo by manipulating its GPS location to a_{init} that satisfies the following equation:

$$
a_{init} = p_{init} + k \cdot (p_{target} - p_{init}), \quad \text{where } k < 0 \text{ and } \|a_{init} - p_{init}\| > (\textit{leash length}). \tag{1}
$$

Equation (1) is the vector form of the parametric equation of a line. k is a free parameter that is allowed to be any negative number; thus, a_{init} lies on the line that passes from p_{init} in the direction from p_{target} to p_{init}. In addition, the distance between a_{init} and p_{init} should be longer than the leash length.

(ii) *Adaptive Spoofing.* The value of fail_count in Algorithm 1 will increase by one if the GPS location of the drone jumps to a_{init}, because the GPS velocity is inconsistent with the velocity measured by the IMUs. The GPS velocity must be changed adaptively to be similar to the motion of the drone body to avoid triggering the EKF fail-safe mode as soon as the GPS location jumps to a_{init} (Figure 9)—a scenario that we call "adaptive spoofing." It makes the GPS location approach the drone's original track to within the leash length, but we can prevent a change in the ITP by manipulating the GPS location to a_{init} again. The hijacker then repeats the process until the drone reaches the safe target location. Note that the EKF failure count increases by one for each jump, but it will decrease to zero if the GPS velocity is consistent with the velocity estimated by the IMUs after each jump. We define t' as the current time and Δ as the GPS location update period. After the fake location jumps to a_{init} again, the current fake location, $a_{t=t'}$, should be altered by changing the physical location during the time Δ, $p_{t=t'} - p_{t=(t'-\Delta)}$, from the last fake location, $a_{t=(t'-\Delta)}$. It can be represented using the following equation,

$$a_{t=t'} = a_{t=(t'-\Delta)} + (p_{t=t'} - p_{t=(t'-\Delta)}).$$

The hijacker can predict the drone's flight path according to Section 7.3.1 and measure the drone's speed in advance. Thus, the hijacker can generate the corresponding spoofing signal in advance to compensate for the propagation delay of the spoofing signals.

(iii) *Maximal Hopping Distance.* Each GPS receiver has its own operational limits in terms of the navigation update rate, acceleration, and velocity. GPS spoofing can cause loss-of-lock and activation of the fail-safe if the manipulated GPS location is abruptly changed and the GPS velocity and acceleration exceed their limits. Thus, the manipulated GPS location should be updated within a range that does not violate the operational limits. We define "maximal hopping distance" as the maximal value of the range. The hijacker should consider maximal hopping distance to avoid activating the fail-safe. In the case of 3DR Solo, it is equipped with the U-blox Neo-7N GNSS module [57]. According to the GPS performance described in its datasheet [56], the maximal navigation update rate is 10Hz, and the operational limit of its velocity is 500m/s. The receiver updates its location every 0.1s when the navigation update rate is 10Hz; thus, the change of the manipulated location is allowed within 50m (i.e., "update period" × "velocity limit of the target receiver"). In other words, the maximal hopping distance of the GPS receiver of 3DR Solo is 50m, and the manipulated GPS location should be updated in this maximal hopping distance.

Depending on the hijacking direction, the distance between a_{init} and p_{init} can be more than the maximal hopping distance, so a hijacker cannot manipulate the GPS location from p_{init} to a_{init} at once. Thus, in this case, the manipulated location should hop multiple times. The EKF failure count also increases while it hops. Therefore, the hijacker should keep the count under 10 by decreasing the count through adaptive spoofing, as described in Figure 9, to avoid activating the fail-safe.

7.3.3 SITL Simulation Results. ArduCopter supports the SITL simulator for developers to execute and test its flight code on their computer without drone hardware. Developers can test their custom flight codes or simulate various events such as wind effects, drone body vibrations, and GPS failures. To execute ArduCopter codes without drone hardware, the SITL simulator supports a physics simulator that simulates the drone's motion using the motor output from the ArduCopter code and the output of the simulated sensors, including a GPS receiver. Figure 10 shows that the

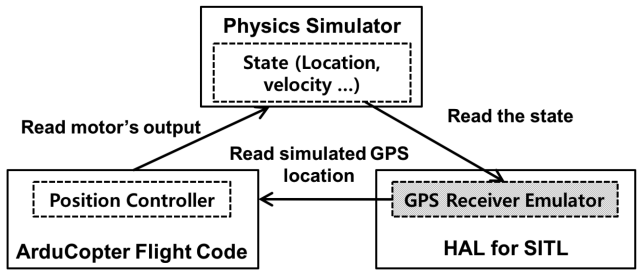


Fig. 10. SITL architecture. We modified the simulated GPS receiver (the gray box) in the hardware abstraction layer (HAL) for SITL.

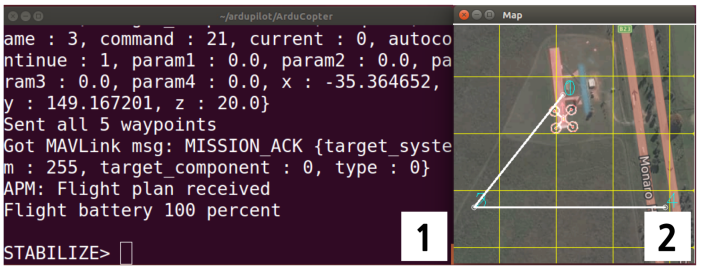


Fig. 11. Screenshot of the SITL: (1) a command prompt at which a user enters commands to SITL, and (2) a map that shows the current position of the simulated drone.

GPS receiver emulator reads the location and velocity of the simulated drone from the state of the physics simulator, and sends new GPS information to the simulated drone.

Before starting the simulation, we modified the GPS emulator to simulate soft GPS spoofing and checked whether our safe-hijacking strategy can successful hijack ArduCopter. We particularly modified the function `_update_gps` in `sitl_gps.cpp` of the hardware abstraction layer (HAL) for SITL. This function is responsible for sending updated GPS information, including the drone's position. Therefore, we replaced the drone's position with the fake location a_{init} obtained from Section 7.3.2 and changed the fake location $a_{t=\Delta}$ (Section 7.3.2). The simulation displays two windows on startup (Figure 11). We uploaded a mission through Window 1 and observed the position and status of the simulated drone through Window 2.

Figure 12 shows an example of safe-hijacking by using the SITL simulator. Points 1 and 2 are the first and second waypoints, respectively. In this simulation, the hijacker wants to move the drone to other points without passing the waypoints.

We simulated the hijacking of a drone for 15 initial fake locations in Figure 13(a). The circular, triangular, and rectangular markers are in zones 1, 2, and 3 in Figure 8, respectively. We configured the takeoff point as (149.17°, −35.36°) and a waypoint as (149.16°, −35.36°) as the longitude and latitude, respectively. On arrival of the simulated drone at the ITP in Figure 13(a), we started safe-hijacking and measured the angular error, which is the difference between the angle of the expected hijacking direction and the angle of the actual moving direction during safe-hijacking. Figure 13(b) presents the box plot of the angular errors according to the zones (Figure 8). The angular error values of zones 1 and 3 were less than 2° and 4°, respectively, in the simulations. The angular error values of zone 2 were less than 9°, but greater than those of zones 1 and 3. This is because the ITP changed when the initial fake location was in zones 2 and 3. This position changed for a longer period in the case of zone 2 than for zone 3, according to Section 7.3.1.

Fig. 12. Points 1 and 2 represent the first and second waypoints. The left figure shows the drone moving to point 1. When safe-hijacking starts, the drone moves southward, not passing point 1, as shown in the middle figure (the upper left green figure describes the fabricated GPS position). At this moment, the initial fake location is located in zone 1. Finally, the right figure shows that the drone has been successfully hijacked.

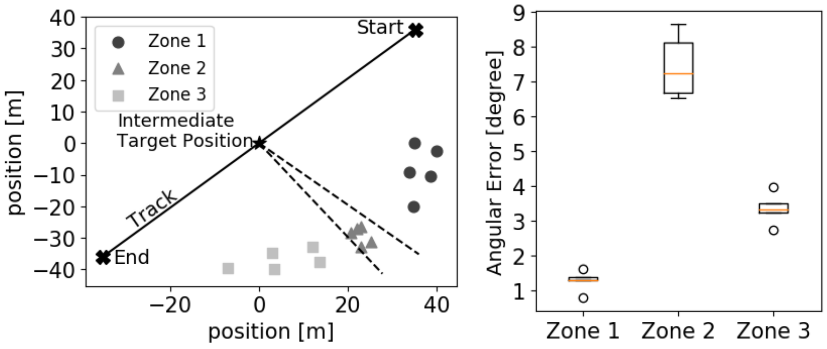


Fig. 13. (a) Simulated environment setup. Circular, triangular, and rectangular markers indicate initial fake locations. (b) Box plot of the angular errors corresponding to each zone containing the initial fake location.

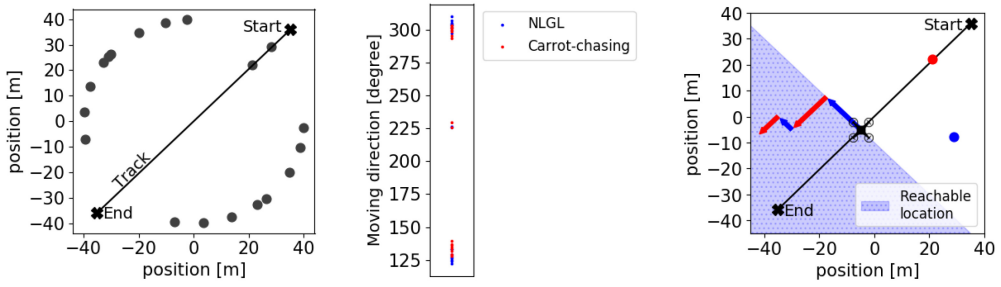
8 DISCUSSION

8.1 Safe-hijacking of Other Consumer Drones

Section 6 presented three safe-hijacking strategies for various consumer drone types, and Section 7 showed that four existing consumer drones can safely be hijacked by applying those strategies. In this subsection, we introduce other path-following algorithms and GPS fail-safe activation mechanisms that consumer drones can adopt, and discuss how our strategies can be applied to the drones that adopt one of them.

8.1.1 Path-following Algorithm. Type I drones can be safely hijacked by using strategy A, regardless of their path-following algorithm in any direction, but the possible hijacking directions of the other types of drones highly depends on their path-following algorithm. In 2014, Sujit et al. investigated and summarized the path-following algorithms widely used in commercial UAVs according to their accuracy, simplicity, robustness, and ease of implementation [51]. From among these algorithms, we applied the carrot-chasing [38] and nonlinear guidance law (NLGL) [39] algorithms—which are suitable for multi-rotor consumer drones—to ArduCopter by modifying its source code, investigated the possible hijacking direction of the drones that employ those algorithms, and simulated it using the SITL simulator.

The carrot-chasing algorithm is similar to that of ArduCopter in that it sets an ITP on the path. The difference lies in the method used to determine the ITP. The carrot-chasing algorithm



(a) (b) (c)

Fig. 14. (a) Simulated environment setup. The pre-planned path is same as that of the simulation in Section 7.3.3. Circular markers indicate the initial fake locations. (b) Moving directions of the simulated drones for various initial fake locations. (c) The drone moves in the direction of the blue arrows or the red arrows when its GPS location is manipulated to the blue dot or the red dot, respectively. The shaded area is reachable by the zigzag movement.

calculates the perpendicular foot from its GPS position to the path, and it determines the ITP that advances a little from the perpendicular foot along the path. Thus, drones that use the carrot-chasing algorithm always have to move in an almost perpendicular direction to the path when they deviate from the path. The NLGL also sets an ITP on the path—it draws a circle around its GPS position and determines the ITP based on the intersection of the circle and the path as ArduCopter does. The difference is that the NLGL makes the drone move in the perpendicular direction when it deviates from the path by more than a certain distance.

We simulated the drone with those algorithms through SITL and applied Strategy C, with the same takeoff point and waypoint configured as in the previous simulation in Section 7.3.3. We manipulated the GPS location into 20 different initial fake locations, as depicted in Figure 14(a), and measured the moving direction of the simulated drone. As a result, the drones move in one of the perpendicular directions to that of the track when the fake location deviates from the track, and they move in the direction of the track when the fake location is on the track. Thus, regardless of the initial fake location, the moving direction converges in three directions: the direction of 315°, the direction of 135°, or the direction of 225°. The results are illustrated in Figure 14(b). Even though there is a restriction on the moving direction when a drone adopts one of these algorithms, the drones can still be hijacked to various locations, because they can zigzag in a series of the perpendicular directions and direction of the track. Figure 14(c) illustrates the locations where the drones can reach by this zigzag movement.

8.1.2 GPS Fail-safe Activation Mechanism. Type I and II drones can be safely hijacked, regardless of their GPS fail-safe activation mechanism, as long as the hijacker can make them lock onto the GPS spoofing signal. However, to hijack other types of drones via GPS spoofing, the hijacker must be able to manipulate their GPS location without activating the GPS fail-safe mode. In the case of 3DR Solo, even though it keeps monitoring the consistency between GPS and IMUs, it does not activate the fail-safe mode until fail_count exceeds 10. Due to this tolerance, 3DR Solo can be hijacked safely, but other drones can adopt different GPS fail-safe activation mechanisms, and Strategy C will not be applicable if the GPS fail-safe mode is activated immediately upon detecting GPS spoofing. However, in order for terrorists to accomplish their missions reliably, their drones must be somewhat tolerant of GPS spoofing detection alarms, as in the case of 3DR Solo.

Numerous spoofing detection methods have been reported in the literature. For instance, Kerns et al. introduced three spoofing detection methods applicable to consumer drones: J/N monitoring,

frequency unlock monitoring, and innovations testing [31]. J/N monitoring involves continuously monitoring the signal and detecting anomalies such as abnormal signal power. Frequency unlock monitoring entails monitoring the loss-of-lock of the GPS signal, as in the case of DJI drones. Innovation testing involves checking the consistency between GPS and IMU outputs, as in the case of 3DR Solo.

Consumer drones can adopt one of these GPS spoofing detection methods, but these drones encounter numerous false alarms because of multipath effects, especially in urban canyons. If the GPS fail-safe mode is activated as soon as GPS spoofing is detected, then the drones will be prevented from experiencing GPS spoofing, but in the case of a false alarm, the drones will be interrupted in normal conditions without GPS spoofing. Therefore, to avoid interfering with their terrorism missions by needlessly triggering the fail-safe mode, the GPS fail-safe activation condition should be relaxed, as in the case of 3DR Solo. Thus, Strategy C will still be applicable to other drones even if the drones adopt other mechanisms that we have not discussed.

8.2 Mitigation of GPS Spoofing Threats to Legitimate Consumer Drones

Our safe-hijacking strategies are primarily a defensive measure against hostile drones. However, safe-hijacking can be maliciously exploited to hijack legitimate consumer drones, and it can be a significant threat to many civilian applications utilizing drones, such as goods delivery, surveillance, and photography. Various spoofing mitigation methods have been proposed. They include suppression of the spoofing signal through digital signal processing (DSP) and nullification of the spoofing signal through multiantenna beamforming [29], but they have several problems when applied to consumer drones. Suppression of the spoofing signal through the DSP approach will not work if the spoofing signal is very strong and the authentic signal is below the sensitivity level of the DSP system. Nullification of the spoofing signal requires both a sufficient number of antennas and sufficient distance between antennas. However, this is not possible as the space available on consumer drones is insufficient for installing multiple antennas, because their motor-to-motor distance is less than 1m in most cases. Therefore, we suggest that an infrastructure that manages drone traffic is required to detect abnormal drone behavior and control them directly in an emergency to mitigate this threat. The National Aeronautics and Space Administration (NASA), the Federal Aviation Administration (FAA), and companies such as Amazon, Boeing, and Google have already begun developing an Unmanned aerial system Traffic Management (UTM) system. Under this system, drones will register their trajectory to the UTM before their flight, and the UTM will track every moving drone in low-altitude airspace. If one of the drones moves in a different trajectory from its registered trajectory, then the UTM will determine that something is wrong with that drone and directly control it to prevent hijacking through GPS spoofing. Further, a GPS threats detection system can be integrated into the UTM, enabling the UTM to provide GPS jamming and spoofing avoidance services. There are several effective spoofing detection methods based on multiple stationary GPS receivers such as the spoofing detection solution for power grids proposed by Yu et al. [62]. By installing multiple GPS receivers on the ground stations of the UTM system, the UTM system will successfully detect GPS spoofing threats and make drones activate GPS fail-safe or change their trajectory to avoid the threat.

8.3 Legal and Safety Issues on GPS Spoofing

In general, it is illegal to generate wireless interference signals, but some countries restrictively allow it on special purposes. For example, federal authorities, including the United States Secret Service, operate equipment that can jam cellphones and other wireless devices to prevent terrorism [54]. In Sweden, the use of radio jammers is allowed in the armed forces and prisons [42]. Seventy units of "DroneGun Tactical" product, equipped with GNSS jammer, were already

exported to a middle eastern country and the company is waiting for US approval soon. This means that increasingly more countries are trying to adopt radio interference, worrying about drone threats [35]. Thus, we believe that safe-hijacking via GPS spoofing could be allowed to guard critical infrastructure if it turns out that GPS spoofing is an effective way to disrupt hostile drones and its interference could be minimized.

Further, the impact on other GPS-dependent systems can be minimized in the following ways. First, the strength and direction of the GPS spoofing signal can be adjusted to minimize the impact. We assume that a drone detection system is utilized to localize the intruding drones for safe-hijacking, so the signal strength can be adjusted to such an extent that it only affects the area very close to the target drone. In addition, a highly directional antenna can be utilized to reduce the impact on surrounding systems. Second, the impact on applications that require timing information through GPS such as power grids and communication systems can be minimized by generating the GPS spoofing signal synchronized to a precise clock such as a GPS disciplined oscillator (GPSDO). A GPSDO maintains its timing accuracy for a while if it has locked onto GPS satellite signals for a long enough time, so this spoofing signal synchronized with authentic GPS time will not affect these applications. Third, in the case of airports, there will be no impact on aircraft if the air traffic control tower makes them stay above the maximum range of the GPS spoofer. As discussed in Section 1, airports have already halt landing and takeoff when drone activity is detected. Thus, the air traffic control tower can make all aircraft awaiting landing avoid GPS spoofing threats by giving them control instructions to wait above a certain altitude.

9 CONCLUSION

The anti-drone market is expected to grow at an annual rate of 23.89% between 2017 and 2022 and it is predicted that it will be worth 1.14 billion USD by 2022 [34]. However, existing anti-drone techniques cannot adequately counteract the threat of malicious consumer drones transporting dangerous materials, such as explosives, because most of these techniques only focus on disrupting or disabling the flight of drones, thus leading to collateral damage. In this study, we analyzed the fail-safe mechanisms of four popular consumer drones via white- and black-box analyses, and developed a taxonomy of consumer drones according to their fail-safe mechanisms. Based on the taxonomy, we developed safe-hijacking strategies for each drone type, and demonstrated the efficacy of these strategies through field experiments and SITL simulations. We expect that anti-drone solutions will equip themselves with a GPS spoofer and envision that our safe-hijacking strategies will be used as effective and safe anti-drone mechanisms in the future.

REFERENCES

[1] 3DR. 2018. 3DR Solo Drone. Retrieved from https://3dr.com/solo-drone/.
[2] ArduPilot Dev Team. 2019. Copter Home—Copter documentation. Retrieved from http://ardupilot.org/copter/.
[3] Battelle Memorial Institute. 2018. Counter-UAS technologies. Retrieved from https://www.battelle.org/government-offerings/national-security/aerospace-systems/counter-UAS-technologies.
[4] BBC. 2015. Japan radioactive drone: Tokyo police arrest man. Retrieved from http://www.bbc.com/news/world-asia-32465624.
[5] BBC. 2017. Drone causes Gatwick airport disruption. Retrieved from https://www.bbc.com/news/amp/uk-40476264.
[6] Kai Borre, Dennis M. Akos, Nicolaj Bertelsen, Peter Rinder, and Søren Holdt Jensen. 2007. *A Software-Defined GPS and Galileo Receiver: A Single-Frequency Approach*. Birkhäuser, New York, NY.
[7] Wenxin Chen, Yingfei Dong, and Zhenhai Duan. 2018. Attacking altitude estimation in drone navigation. In *Proceedings of the IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS'18)*. IEEE, 888–893.
[8] Wenxin Chen, Yingfei Dong, and Zhenhai Duan. 2018. Manipulating drone dynamic state estimation to compromise navigation. In *Proceedings of the 2018 IEEE Conference on Communications and Network Security (CNS'18)*. IEEE, 1–9.

[9] James R. Clynch. 2006. Geodetic coordinate conversions I. Retrieved from https://www.oc.nps.edu/oc2902w/coord/coordcvt.pdf.
[10] CNN. 2019. Dubai airport temporarily halts flights due to "drone activity." Retrieved from https://edition.cnn.com/2019/02/15/middleeast/dubai-airport-drone-intl/index.html.
[11] Youjing Cui and Shuzhi Sam Ge. 2003. Autonomous vehicle positioning with GPS in urban canyon environments. *IEEE Trans. Robot. Autom.* 19, 1 (2003), 15–25.
[12] Dedrone. 2018. DroneTracker now features an integrated jammer. Retrieved from https://www.dedrone.com/press/dronetracker-now-features-an-integrated-jammer.
[13] Dedrone. 2018. DroneTracker software. Retrieved from https://www.dedrone.com/products/software.
[14] Dedrone. 2018. Event kit overview. Retrieved from https://www.dedrone.com/products/hardware/event-kit/event-kit-overview.
[15] DJI. 2015. *Phantom 3 Standard User Manual.* Retrieved from http://dl.djicdn.com/downloads/phantom_3_standard/en/Phantom_3_Standard_User_Manual__V1.4.pdf.
[16] DJI. 2016. *Phantom 4 User Manual.* Retrieved from https://dl.djicdn.com/downloads/phantom_4/en/Phantom_4_User_Manual_en_v1.0.pdf.
[17] DJI. 2017. DJI and UNDP use latest drone technology to protect vulnerable communities. Retrieved from http://citizenship.dji.com/humanitarian/dji-and-undp-use-latest-drone-technology-to-protect-vulnerable-communities.
[18] DJI. 2019. Phantom 3 Standard—Drone for beginners. Retrieved from https://www.dji.com/phantom-3-standard.
[19] DJI. 2019. Phantom 4—DJI's smartest flying camera ever. Retrieved from https://www.dji.com/phantom-4.
[20] Drone Defence. 2019. NetGun X1 | short range drone protection. Retrieved from https://www.dronedefence.co.uk/products/netgun-x1.
[21] DroneShield. 2018. DroneSentry. Retrieved from https://www.droneshield.com/sentry/.
[22] DroneShield. 2018. DroneShield. Retrieved from https://www.droneshield.com/.
[23] DroneShield. 2018. How DroneShield works. Retrieved from https://www.droneshield.com/how-droneshield-works/.
[24] Express Newspapers. 2017. Laser weapon called the "Silent Hunter" that can slice MISSILES unveiled by China. Retrieved from https://www.express.co.uk/news/uk/774268/Defence-China-US-Laser-Weapon-Tank-Slicing-Defence-Abu-Dhabi.
[25] Daniele Giordan, Yuichi S Hayakawa, Francesco Nex, and Paolo Tarolli. 2018. Preface: The use of remotely piloted aircraft systems (RPAS) in monitoring applications and management of natural hazards. *Natural Hazards and Earth System Sciences* 18, 11 (2018), 3085–3087.
[26] GIS Lounge. 2015. Use of drones in GIS. Retrieved from https://www.gislounge.com/use-drones-gis/.
[27] Lin Huang and Qing Yang. 2015. Low-cost GPS Simulator - GPS Spoofing by SDR. https://media.defcon.org/DEF%20CON%2023/DEF%20CON%2023%20presentations/DEFCON-23-Lin-Huang-Qing-Yang-GPS-Spoofing.pdf
[28] Todd E. Humphreys, Brent M. Ledvina, Mark L. Psiaki, Brady W. O'Hanlon, and Paul M. Kintner Jr. 2008. Assessing the spoofing threat: Development of a portable GPS civilian spoofer. In *Proceedings of the ION GNSS International Technical Meeting of the Satellite Division*, Vol. 55. ION, Manassas, VA, 2314–2325.
[29] Ali Jafarnia-Jahromi, Ali Broumandan, John Nielsen, and Gérard Lachapelle. 2012. GPS vulnerability to spoofing threats and a review of antispoofing techniques. *Int. J. Nav. Obs.* 2012 (2012). https://www.hindawi.com/journals/ijno/2012/127072/abs/.
[30] Jammerall. 2019. Portable 30 meter 6 Antenna GPS L1 L2 L3 L4 L5 Glonass all bands GPS signal jammer and Lojack WiFi jammer. Retrieved from http://www.jammerall.com/products/Portable-30-meters-6-Antenna-GPS-L1-L2-L3-L4-L5-Glonass-All-Bands-GPS-Signal-Jammer-%26-Lojack-WiFi-Jammer.html.
[31] Andrew J. Kerns, Daniel P. Shepard, Jahshan A. Bhatti, and Todd E. Humphreys. 2014. Unmanned aircraft capture and control via GPS spoofing. *J. Field Robot.* 31, 4 (2014), 617–636.
[32] Aaron Luo. 2016. Drones Hijacking - Multi-dimensional Attack Vectors and Countermeasures. https://media.defcon.org/DEF%20CON%2024/DEF%20CON%2024%20presentations/DEFCON-24-Aaron-Luo-Drones-Hijacking-Multi-Dimensional-Attack-Vectors-And-Countermeasures-UPDATED.pdf
[33] Juliette Marais, Marion Berbineau, and Marc Heddebaut. 2005. Land mobile GNSS availability and multipath evaluation tool. *IEEE Trans. Vehic. Technol.* 54, 5 (2005), 1697–1704.
[34] MarketsandMarkets Research. 2016. Anti-drone Market worth 1.14 Billion USD by 2022. Retrieved from http://www.marketsandmarkets.com/PressReleases/anti-drone.asp.
[35] My Security Media. 2018. 70 Australian-made DroneGun Tactical anti-drone devices sold to middle Eastern Country. Retrieved from http://drasticnews.com/70-australian-made-dronegun-tactical-anti-drone-devices-sold-to-middle-eastern-country/.
[36] OpenWorks Engineering. 2018. SkyWall. Retrieved from https://openworksengineering.com/skywall.
[37] Orolia. 2017. GPS and GNSS satellite simulators—GSG series. Retrieved from https://www.orolia.com/products-services/gnss-simulation/gpsgnss-simulators.

[38] Paparazzi Autopilot Team. 2017. Paparazzi is a free and open-source hardware and software project for unmanned (air) vehicles. Retrieved from https://github.com/paparazzi/paparazzi.
[39] Sanghyuk Park, John Deyst, and Jonathan P. How. 2007. Performance and lyapunov stability of a nonlinear path following guidance method. *J. Guid. Contr. Dynam.* 30, 6 (2007), 1718–1728.
[40] Parrot. 2016. Parrot Bebop 2 Drone User Guide. Retrieved from https://parrotcontact.parrot.com/website/user-guides/download-user-guides.php?pdf=bebop-2/Bebop-2_User-guide_UK.pdf.
[41] Parrot. 2019. Parrot Bebop 2 drone. Retrieved from https://www.parrot.com/us/drones/parrot-bebop-2.
[42] PTS. 2015. Förbud mot störsändare. Retrieved from https://pts.se/sv/privat/radio/utrustning/forbud-mot-storsandare.
[43] RACELOGIC. 2018. Labsat 3 GPS Simulator. Retrieved from https://www.labsat.co.uk/index.php/en/products/labsat-3.
[44] Aanjhan Ranganathan, Hildur Ólafsdóttir, and Srdjan Capkun. 2016. SPREE: A spoofing resistant GPS receiver. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking (MobiCom'16)*. ACM, New York, NY, 348–360. DOI: https://doi.org/10.1145/2973750.2973753
[45] Reuters. 2016. Dubai airport shuts airspace for nearly 30 mins due to drone activity. Retrieved from https://www.reuters.com/article/us-emirates-dubai-airport-drone-idUSKCN11Y0TY.
[46] Fred Samland, Jana Fruth, Mario Hildebrandt, Tobias Hoppe, and Jana Dittmann. 2012. AR. drone: Security threat analysis and exemplary attack to track persons. In *Proceedings of the SPIE*, Vol. 8301. SPIE, Bellingham, WA, 1–15.
[47] Singapore Press. 2017. Drones disrupt over 240 flights in Chongqing. Retrieved from https://www.straitstimes.com/asia/east-asia/drones-disrupt-over-240-flights-in-chongqing.
[48] Yunmok Son, Hocheol Shin, Dongkwan Kim, Youngseok Park, Juhwan Noh, Kibum Choi, Jungwoo Choi, and Yongdae Kim. 2015. Rocking drones with intentional sound noise on gyroscopic sensors. In *Proceedings of the 24th USENIX Conference on Security Symposium (SEC'15)*. USENIX Association, Berkeley, CA, 881–896.
[49] Spirent. 2018. GSS7000 Series. Multi-GNSS, multi-frequency constellation simulators. Retrieved from https://www.spirent.com/Products/GSS7000.
[50] Stuff. 2018. Drone scare delays Auckland Airport flights, turns out to be balloon. Retrieved from https://www.stuff.co.nz/travel/travel-troubles/103111136/drone-scare-delays-auckland-airport-flights-turns-out-to-be-balloon.
[51] P. B. Sujit, Srikanth Saripalli, and Joao Borges Sousa. 2014. Unmanned aerial vehicle path following: A survey and analysis of algorithms for fixed-wing unmanned aerial vehicless. *IEEE Contr. Syst.* 34, 1 (2014), 42–59.
[52] Takuji Ebinuma. 2018. Software-defined GPS signal simulator. Retrieved from https://github.com/osqzss/gps-sdr-sim.
[53] The New York Times. 2016. Pentagon confronts a new threat from ISIS: Exploding drones. Retrieved from https://www.nytimes.com/2016/10/12/world/middleeast/iraq-drones-isis.html?_r=0.
[54] The Washington Post. 2009. Local police want right to jam wireless signals. Retrieved from http://www.washingtonpost.com/wp-dyn/content/article/2009/01/31/AR2009013101548.html?noredirect=on.
[55] Nils Ole Tippenhauer, Christina Pöpper, Kasper Bonne Rasmussen, and Srdjan Capkun. 2011. On the requirements for successful GPS spoofing attacks. In *Proceedings of the 18th ACM Conference on Computer and Communications Security (CCS'11)*. ACM, New York, NY, 75–86. DOI: https://doi.org/10.1145/2046707.2046719
[56] u-blox. 2014. NEO-7 U-blox 7 GNSS modules Data Sheet. Retrieved from https://www.u-blox.com/sites/default/files/products/documents/NEO-7_DataSheet_%28UBX-13003830%29.pdf.
[57] u-blox. 2017. NEO-7 series. Retrieved from https://www.u-blox.com/en/product/neo-7-series.
[58] UNA-UK. 2013. Noel Sharkey on drones and the threat of autonomous weapons. Retrieved from https://www.una.org.uk/magazine/spring-2013/noel-sharkey-drones-and-threat-autonomous-weapons.
[59] Verizon Media. 2016. "Icarus" machine can commandeer a drone mid-flight. Retrieved from https://www.engadget.com/2016/10/28/icarus-hijack-dmsx-drones/.
[60] WIRED. 2015. Welcome to the world, drone-killing laser cannon. Retrieved from https://www.wired.com/2015/08/welcome-world-drone-killing-laser-cannon/.
[61] WIRED. 2018. The explosive-carrying drones in Venezuela won't be the last. Retrieved from https://www.wired.com/story/venezuela-drones-explosives-maduro-threat/.
[62] Der-Yeuan Yu, Aanjhan Ranganathan, Thomas Locher, Srdjan Capkun, and David Basin. 2014. Short paper: Detection of GPS spoofing attacks in power grids. In *Proceedings of the 2014 ACM Conference on Security and Privacy in Wireless & Mobile Networks (WiSec'14)*. ACM, New York, NY, 99–104. DOI: https://doi.org/10.1145/2627393.2627398

Received May 2018; revised October 2018; accepted January 2019