Sensors and Materials, Vol. 32, No. 8 (2020) 2729–2743
MYU Tokyo

S & M 2296

Hijacking Unmanned Aerial Vehicle by Exploiting Civil GPS Vulnerabilities Using Software-defined Radio

Xian-Chun Zheng and Hung-Min Sun*

Department of Computer Science and Information Engineering, National Tsing Hua University,
No. 101, Section 2, Guangfu Road, East District, Hsinchu City 300, Taiwan

(Received December 9, 2019; accepted June 12, 2020)

Keywords: SDR, UAV, GPS, hijacking, civil GPS vulnerability

With the booming growth of unmanned aerial vehicles (UAVs, drones) in recent years, especially for commercial and recreational aerial photography UAVs, the safety issues of civilians and security problems of sensitive information are causing much concern. Some studies have confirmed that many attacks against UAVs are highly connected to the vulnerabilities of the civil global positioning system (GPS). The transparency and predictability of unencrypted civil GPS signals make them easy to counterfeit. Furthermore, owing to the development of software-defined radio (SDR) in recent years, launching a civil GPS spoofing attack is no longer expensive. GPS spoofing against drones using SDR devices has severely threatened their flight security and personal safety. In this study, we demonstrated three UAV hijacking attacks using HackRF One, an SDR device, with corresponding open source projects. We also proved that launching such threatening attacks is not expensive and such attacks are easy to control. We also propose some possible approaches to enhance the security of UAV location information.

1. Introduction

With the boom in unmanned aerial vehicles (UAVs, drones) in recent years, especially for commercial and recreational aerial photography UAVs, people are becoming more interested in drones and ownership is increasing. Not only commerce but also academia is becoming more concerned about relevant issues, such as Prime Air, a novel type of order delivery proposed by Amazon, and related safety/security research.[1] The aim of Prime Air is to deliver packages to customers with drones in 30 min or less. We expect to see drone-related applications to spread worldwide in the coming years.

A UAV is defined as "a powered, aerial vehicle that does not carry a human operator, uses aerodynamic forces to provide vehicle lift, can fly autonomously or be piloted remotely, can be expendable or recoverable, and can carry a lethal or nonlethal payload".[2] Additionally, the global positioning system (GPS) plays an important role in stable and trusted flight navigation. However, some studies have confirmed that many attacks against drones are

*Corresponding author: e-mail: hmsun@cs.nthu.edu.tw
https://doi.org/10.18494/SAM.2020.2783

ISSN 0914-4935

highly connected to the vulnerabilities of the civil GPS. Tippenhauer *et al.*[3] used a Spirent GSS7700 GPS simulator to conduct a set of experiments in which they investigated the required precision of successful GPS spoofing attacks. Humphreys[4] claimed that the transparency and predictability of unencrypted civil GPS signals make them easy to counterfeit, and disclosed the vulnerabilities of GPS-related systems.

Recently, owing to the development of software-defined radio (SDR) platforms, launching such civil GPS spoofing attacks is no longer so difficult. Zeng *et al.*[5] proposed a practical real-time GPS spoofing attack in a road navigation scenario implemented on a low-cost spoofer using HackRF One. Successful spoofing attacks can be launched against various location-based applications on multiple devices. In some countries, SDR systems are not regulated commodities and are available for civilians. Furthermore, there are many open source projects [e.g., GPS-SDR-SIM project[6]] on the Internet. A user can launch a simple GPS spoofing attack with merely a few commands. As a result, GPS spoofing attacks will no longer require complicated operations, and this will be a severe threat against UAVs.

1.1 Contribution

In this paper, we present how to launch a GPS spoofing attack against the navigation system of drones utilizing an SDR platform and open source projects. Three approaches to hijacking a flying drone will be demonstrated: (1) forcing the flying drone to land, (2) guiding the flying drone to an attacker-specified area, and (3) forcing the flying drone to land on an attacker-specified area (i.e., the drone will not land at the originally scheduled position). Afterwards, we proposed some possible approaches to enhance the security of a drone's location information.

1.2 Organization

The rest of the paper is organized as follows. In Sect. 2, we review some essential background knowledge on GPS and UAVs. In Sect. 3, we introduce some important characteristics of SDR platforms and the use of relevant open source projects. Afterwards, we give an overview of our experiments and related results in Sect. 4. Conclusions are given in Sect. 5.

2. Background

2.1 GPS

1) *Basis of GPS:* GPS is a global navigation system that uses satellites to provide precise three-dimensional position, velocity, and time information.[7] It consists of 24 satellites in six different orbits. All of these satellites are equipped with high-precision atomic clocks. Every satellite continuously broadcasts a GPS signal, which contains its coordinates and data transmission time, which helps receivers calculate their latitude, longitude, and altitude.

The GPS satellites broadcast two different signals, a civilian unencrypted signal that transmits on the L1 band (1575.42 MHz) and a military encrypted signal that transmits on the

L2 band (1227.60 MHz). The military signal is encrypted to prevent unauthorized use and imitation, and is used by authorized US military receivers only. The civilian signal is not encrypted for civilian access.

2) *GPS Positioning Principle:* GPS uses a number of satellite transmitters *Si* located at known positions x_i, y_i, $z_i \in R^3$ for i in {1, 2, ..., N}, where N is the total number of working satellites, each carrying a high-precision synchronized clock.

We assumed a GPS receiver located at unknown coordinates $\{x_0, y_0, z_0\} \in R^3$. The receiver T uses an omnidirectional antenna to receive positioning signals with a time stamp from all satellites in the range; then, it can calculate all distances to each satellite transmitter from the signal travel duration τ_i since the speed of the electromagnetic wave c is known. However, the receiver's clock is not synchronized with those on satellites, and there is a time difference Δt. Thus, the distance between the receiver T and the GPS satellite transmitter S_i can be denoted by R_i, which is given by

$$R_i = c(\tau_i + \Delta t), \tag{1}$$

and the positioning equation can be written as

$$(x_i - x_0)^2 + (y_i - y_0)^2 + (z_i - z_0)^2 = R_i^2. \tag{2}$$

To solve the equation with four unknowns, the receiver T needs to receive GPS satellite signals from at least four separate transmitters at one time,[8] and it will obtain a set of positioning equations for an overdetermined system. Owing to noise, these equations generally do not have a unique solution. Hence, we can solve this problem by the least-mean-squares approach. The core concept is shown in Fig. 1.

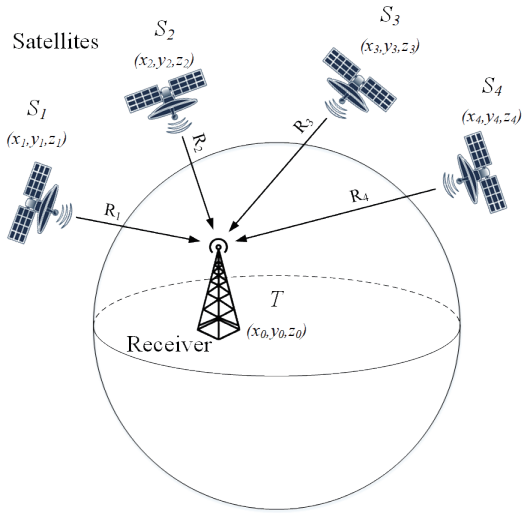

Fig. 1. (Color online) Positioning principle.

3) *Vulnerabilities of Civil GPS:* The main vulnerabilities of civil GPS signals result from their transparency and predictability, the very characteristics that have made civil GPS signals so popular worldwide.[4] Unencrypted GPS signals are transparent to every receiver, and most information about their spreading codes is openly documented on websites;[9] moreover, satellites have regular orbits. This is the reason why the reliable prediction of civil GPS signals is trivial. In addition, transparency and predictability make the civilian GPS signals possible to counterfeit.

2.2 UAVs

1) *Characteristics of Drones:* To ensure the safety and reliability of flight, the vast majority of civil drones require reliable navigation. By far, the most common approach to ensuring reliable drone navigation has been to build a state estimator around a sensor core consisting of an inertial measurement unit (IMU) and a GPS receiver.[10] Most civil drones are equipped with a high-resolution camera for recreational aerial photography. Although many drones are equipped with additional sensors, such as altimeters and an infrared sensing system, GPS is fundamental and crucial to the sensor suite; unlike the other sensors, GPS works under all weather conditions with no drift.

2) *Weaknesses of Drones:* There are some common security issues of civil drones. Many of their characteristics and vulnerabilities are reviewed and discussed in Ref. 11. First, there exists a trade-off between the security of civil drones and their functionalities, expected benefits, and cost. To provide smooth and superior user experience, vendors face the same limitations as those of Internet of Things (IoT) devices: small size, weight constraints, and low computational power.[12] Because drones are complicated aircraft composed of several subsystems, sensors, and long-distance interactions, it is difficult to realize security protection on civilian drones.

Second, as described above, GPS plays an important and now irreplaceable role in drones. In addition to basic positioning and reporting locations on maps, other functions such as *follow me*, *return to home*, and *no-fly zone* are based on GPS. However, the transparency of civil GPS signals has already threatened the security of civil drones.

3. GPS Spoofing Using SDR

3.1 SDR platform

SDR is a radio communication system where components that have been typically implemented in hardware are implemented by means of software on a personal computer or an embedded system.[13] Different from most radio frequency (RF) platforms/devices, an SDR platform can be controlled by programs. That is, programmers do not need to change any configurations on hardware even if they transmit at various RFs. In this study, we utilized an SDR platform to transmit a counterfeit GPS signal for our GPS spoofing experiments. The remaining part of this section introduces HackRF One, the SDR hardware platform we used in our experiments and related open source projects.

3.2 SDR hardware platform

HackRF One is a low-cost open source SDR platform, which costs about $300 USD.[14] It operates over a wide range of frequencies from 1 MHz to 6 GHz, which covers the frequency band of GPS satellite signals. It also supports sample rates up to 20 million samples per second. HackRF One provides a USB 2.0 interface for data transmission and a power supply, a SubMiniature version A (SMA) interface for the antenna, and an SMA interface for the clock input and output. It also has internal pin headers for expansion. The main drawback is that HackRF One only supports half-duplex operation, which means that it cannot transmit and receive signals simultaneously.

In this study, HackRF One is used as a civil GPS spoofer. However, it is ineffective for transmitting counterfeit GPS signals with its default factory packaging. The built-in oscillator of HackRF One has a tolerance of 20 ppm, while it needs an accuracy of at least 1 ppm for GPS signal simulation.[15] An oscillator with low accuracy may cause the transmitting signal to be unacceptable and undecodable for civil GPS receivers. To address this issue, we decided to utilize a temperature-compensated crystal oscillator (TCXO)[16] with a tolerance of 0.1 ppm, which satisfies the requirements listed in Ref. 17, and we installed it as shown in Fig. 2. Using Linux Ubuntu, we can download the command line tool[17] and transmit the signal using the hackrf_transfer tool.

In Fig. 3, HackRF One is transmitting a GPS signal with pregenerated signal data [introduced in detail in Sect. 3.3, 1)]. Transmission is repeated again and again without any interruptions.

3.3 SDR open source projects

A typical GPS spoofing attack operation includes two steps: (1) generating a binary transmitting file and (2) transmitting a GPS signal with the previously generated binary file, where the flowchart is shown in Fig. 4. In Sect. 3.2, we illustrated the second step of the spoofing operation. Afterwards, we demonstrate how to generate the binary transmitting file, the first step of the spoofing operation, using open source projects in the following paragraphs.

Fig. 2. (Color online) HackRF One with TCXO.

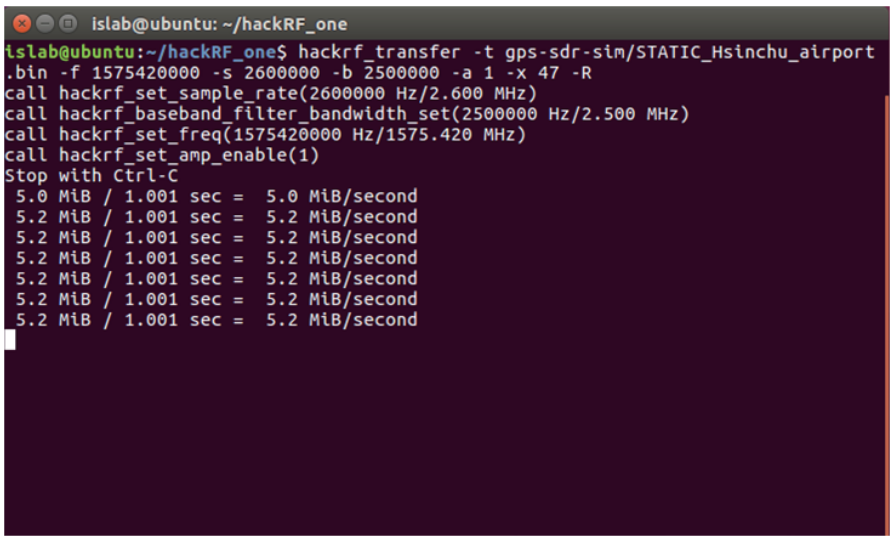

Fig. 3. (Color online) Transmitting GPS signal data.

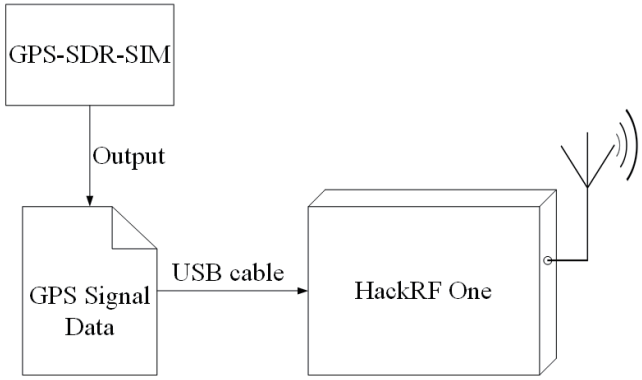

Fig. 4. GPS spoofing mechanism.

1) *GPS-SDR-SIM:* This open source project[6] helps us generate the GPS signal data used by HackRF One in the second step. This project provides us with two different modes of spoofing attack that may be reflected in the user motion at the receiver side: static and dynamic modes. Under the static mode, the spoofed receiver will report a static position during the whole transmission period. In contrast, under the dynamic mode, the spoofed receiver will report motion. To launch dynamic spoofing, we also need to additionally input a user motion file. In our experiments, spoofing with the static mode is sufficient, so we mainly focus on the static mode in this paper.

Before we begin to generate GPS signal data, GPS broadcast ephemeris files with the RINEX format[18] should be downloaded and decompressed, which can be found in Ref. 9. These files contain unique GPS satellite ephemeris messages every day, which allows the GPS receiver to calculate its coordinates through the data provided by the satellite. That is,

this project can generate fake GPS signal data, which causes the GPS receiver to report a false location. To fetch particular ephemeris files, we can append various directory and file names to the starting directory,[19] where these archives are named with specific rules. For example, *brdc0520.17n* represents the ephemeris messages of February 21st, 2017.

In Fig. 5, the GPS signal data is generated in 80 s with a transmission duration of 300 s. Only messages of visible satellites at the target location are calculated and shown in the output.

2) *Spoofer with Joystick:* This open source project[20] contains three subprojects. Here, we only utilize the "Realtime-generate-fake-GPS-by-joystick" project, which is capable of manipulating GPS receivers' reported position by joystick immediately. This project is a modification of the GPS-SDR-SIM project.[6] It provides a compiled program executable on a Windows operating system. In contrast to the GPS-SDR-SIM project, this project provides one-step operation where it combines the two above steps, and it allows us to manipulate the target location immediately by keyboard or joystick. The location reported from the receiver will change as a result, making it seem like a victim is moving on Google Maps after the spoofer's commands. The command line user interface is shown in Fig. 6, and the attacker can easily modify the latitude and longitude by a key in W/A/S/D (direction operation) or by joystick. However, there is no interface for user motion input to launch dynamic spoofing.

4. Experiments

4.1 Design

The main goals of our experiments are listed below, and details of the processes carried out to achieve each goal are described in the following subsections.

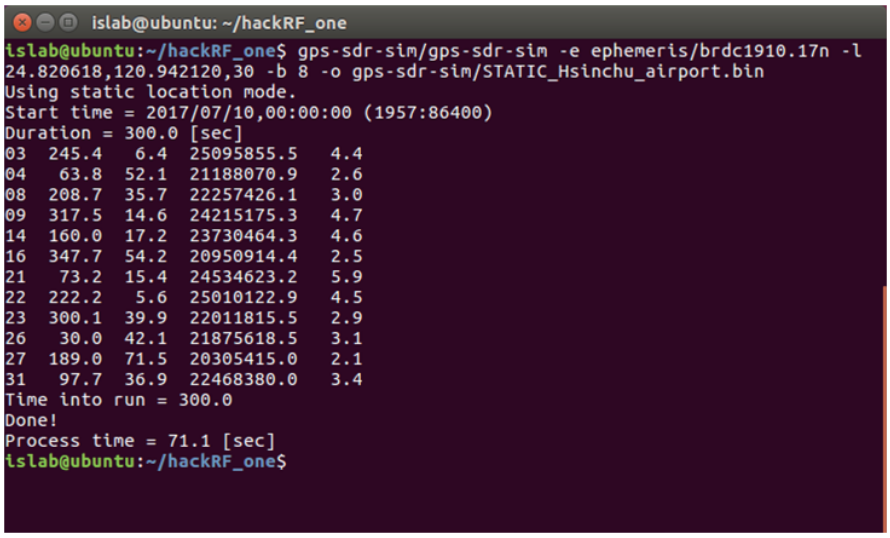

Fig. 5. (Color online) Generating GPS signal data.

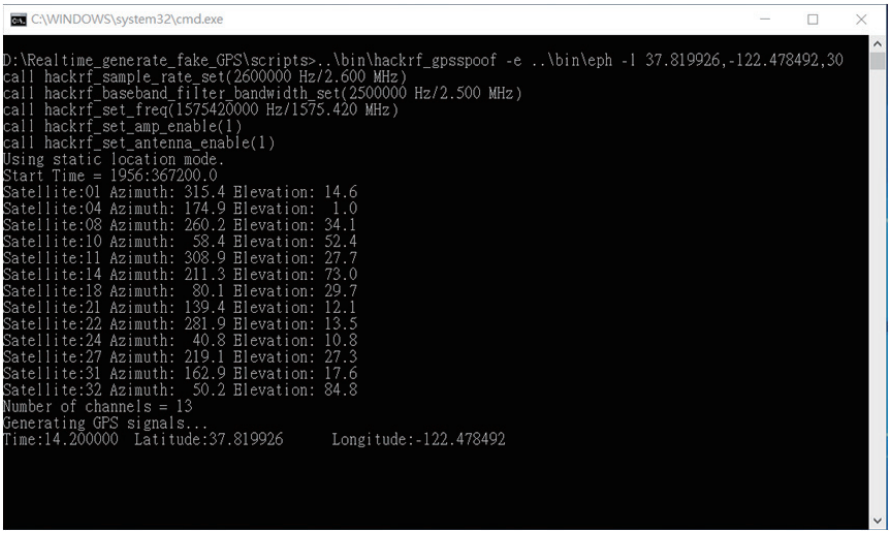

Fig. 6. Spoofing with joystick.

1) To force the flying drone to land
2) To guide the flying drone to fly in an attacker-specified direction
3) To force the flying drone to land on an attacker-specified area

1) *Forced Landing:* Generally, there are some protection mechanisms for a flying drone that encounters inevitable problems, such as losing control messages from the remote controller and losing the GPS signal. Different strategies are adopted by different vendors/developers, and most of the time, the drone will land at the current location or return to the scheduled position, mainly depending on the problem.

To force a drone to land immediately, we will exploit the vulnerability of a no-fly zone. A no-fly zone is a range of airspace in which no aircraft is permitted to fly. Modern recreational aerial photography drones have built-in maps attached with no-fly zones that they avoid or leave when flying into them. A detailed flowchart of forced landing attack is shown in Fig. 7. We will transmit a fake GPS signal when the drone is hovering. After the repositioning of the drone, the drone will be reported to the location at Hsinchu Airport, a no-fly zone in Hsinchu City.

2) *False Direction Guiding:* Many core features and interesting services of drones are highly dependent on the GPS service, such as stable hovering and flying, navigation, protection mechanisms, flight records and logs, and so on, although skilled users can fly in an advanced mode, which provides faster flying and quicker reactions of drones.

Here, we exploit the vulnerability of the stable hovering mechanism, which strongly depends on GPS to help the drone fly at a fixed location when hovering. Without any control message from the remote controller, the drone should not show any movement or location drift. In Fig. 8, if the drone's reported location moves from (x, y) to (x', y'), the stable hovering mechanism will automatically move the drone back to its original position (x, y), which we call a flying adjustment. This phenomenon can be observed when the drone hovers in windy airspace.

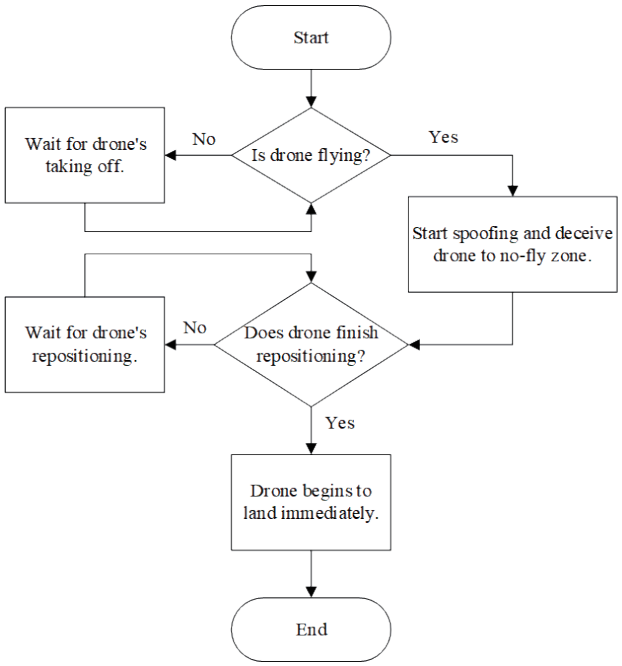

Fig. 7. Flowchart of forced landing attack.

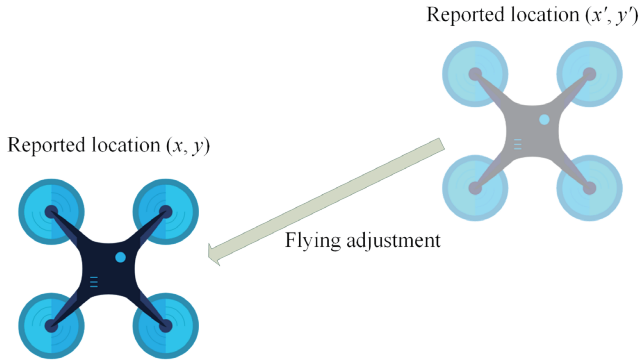

Fig. 8. (Color online) Stable hovering mechanism.

Consequently, we will use the spoofer to change its reported location. This will make the drone fly in a different direction.

Figure 9 demonstrates how we launch a false direction guiding attack. Basically, we can only deceive those drones that fly with the aid of GPS.

3) *Landing on False Area:* To achieve this goal, we exploit the vulnerability of the *go home* function. Firstly, we have to force the drone to *go home* by jamming its communication channel. In this case, the drone will fly back to where it took off or another scheduled position, which varies among different vendors/developers. Secondly, we will spoof the flying drone to a particular place instead of its original destination. We can spoof it with the false direction guiding approach mentioned in Sect. 4.1, 2). A flowchart of landing on a false area is shown in Fig. 10.

4.2 Setup

We use a DJI Phantom 3 Standard aerial photography drone[21] as our target machine in the overall experiments. As with most aerial photography drones, the Phantom 3 Standard is equipped with GPS and a no-fly zone regulation. There are three different flight modes: P-Mode (positioning), A-Mode (attitude), and F-Mode (function). P-Mode provides fixed hovering and stable flying with the assistance of GPS, A-Mode provides faster flying and more agile reactions without any assistance of GPS, and F-Mode provides many additional functions that help the user perform some actions with less effort, including *Follow Me*, *Waypoints*, and so on. Most functions in F-Mode cannot be performed without the assistance of GPS.

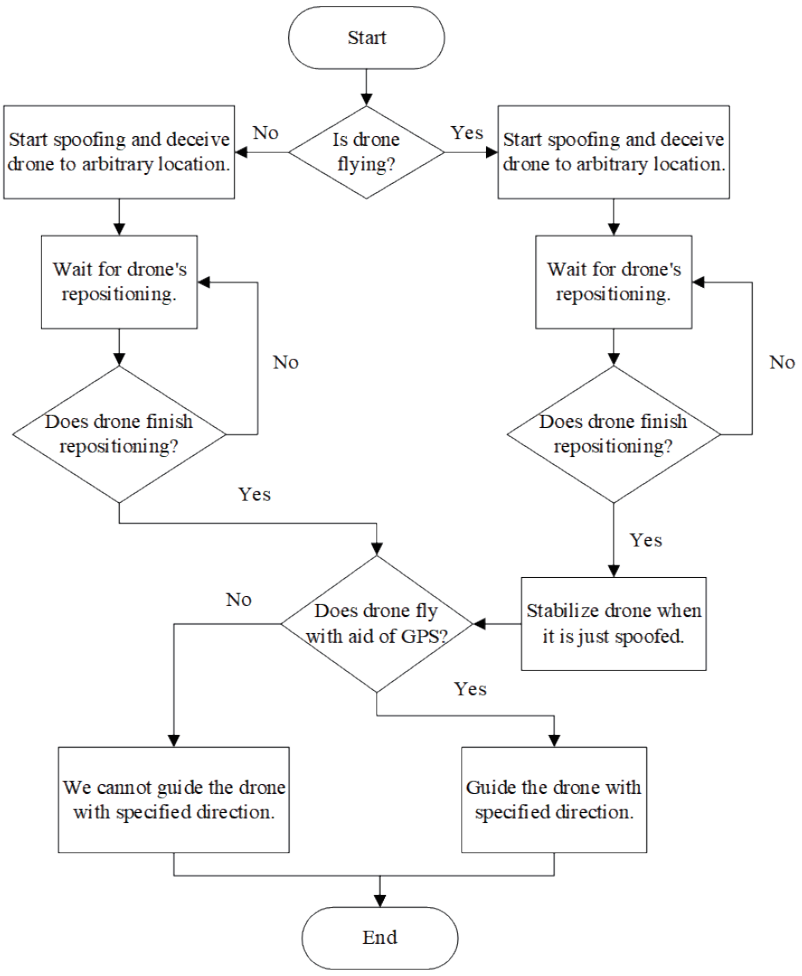

Fig. 9. Flowchart of false direction guiding attack.

Every user has to install the DJI GO app[22] on his/her mobile device to establish a secure connection to the drone. Actually, the remote controller is designed as a special access point (AP). It communicates with the user's mobile device on the 2.4 GHz Industrial Scientific Medical (ISM) band and with the drone on the 5.725 to 5.825 GHz band.

We also use a host laptop to interact with HackRF One, where the laptop has an Intel i5-6198DU CPU and 8 GB of RAM, and runs on Windows 10 (64-bit). To conduct experiments with the GPS-SDR-SIM project, we create a virtual machine (VM) on the host laptop, which runs Ubuntu 16.04 LTS (64-bit) with two processor cores and 2 GB of RAM.

5. Results and Discussion

1) *Forced Landing:* In this experiment, we conducted two different types of tests: (1) emitting a fake GPS signal before the drone took off and (2) emitting a fake GPS signal while it is flying in air. The spoofing target locations in both tests are the same, Hsinchu Airport. We

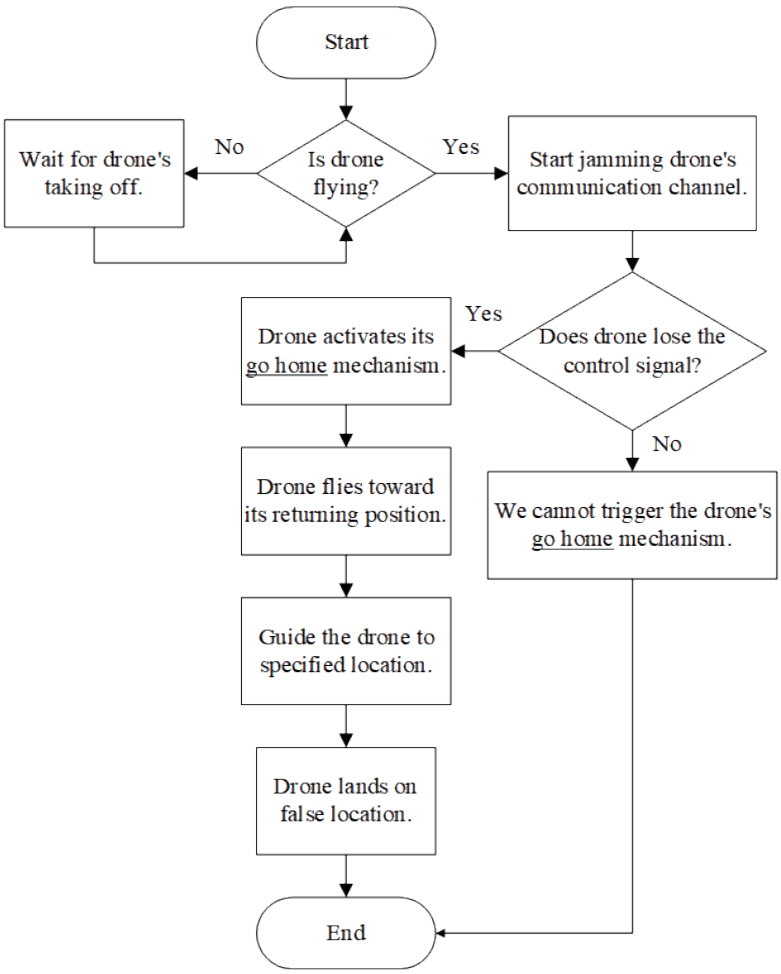

Fig. 10. Flowchart of landing on false area.

Fig. 11. (Color online) Flight restriction.

can see the result of test (1) in Fig. 11, which shows a screenshot from the mobile device. As the drone reported its location to be in the no-fly zone, it could not take off regardless of the flight mode.

In test (2), we started to transmit a counterfeit GPS signal after the drone took off and stably hovered in air. As the false GPS signal became stronger, the GPS receiver began to lose the trace of the original real GPS signal and to receive the fake GPS signal, causing it to recalculate its coordinates. This took about 20 to 40 s; however, it took about 30 to 90 s using the mobile device. Once the drone reported its location in the no-fly zone, it immediately began to land, as predicted (Fig. 12). In Fig. 13, we can clearly see that the drone was spoofed to Hsinchu Airport. In our experiments, we forced the drone to land in 2 min (repositioning time) with a success rate of about 90%.

2) *False Direction Guiding:* We conducted our experiments on the sports ground of Tsing Hua University. We stood at the center of the playground with the drone flying above the playground. Our attacking flight plan is demonstrated in Fig. 14. At the beginning, the drone was hovering at the grey drone icon (real position), and its location was reported (false reported position) as the red aircraft icon. Our aim was to guide the drone to fly along the orange line, ending at the northeast of the playground.

Fig. 12. (Color online) Screenshot of forced landing demo video.

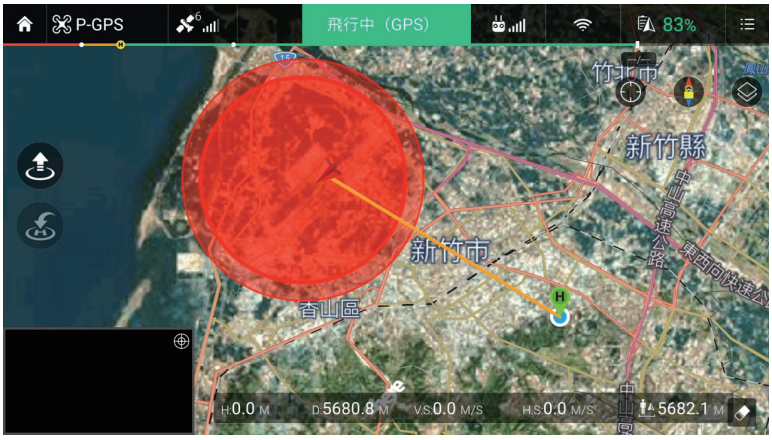

Fig. 13. (Color online) False location at Hsinchu airport.

Fig. 14. (Color online) Attacking flight plan.

Fig. 15. (Color online) Result of false direction guiding.

Based on the stable hovering mechanism, we used the spoofer with the joystick project and changed the drone's reported location, making it move a short distance to the west. The drone flew to the east, that is, the opposite direction. The result is shown in Fig. 15, where spoofing the drone with one direction is defined as an operation. In our experiments, about 80% of the operations succeeded. Flying out of range and the imprecise positioning of the drone's GPS receiver may be the reasons for the failed operations.

3) *Landing on False Area:* In this experiment, we planned to launch a jamming attack on the control signal band with HackRF One to force the drone to *go home*. Different drones operate on different RF bands. The DJI Phantom 3 Standard transmits its control messages on the 5.725 to 5.825 GHz band. We analyzed the spectrum in this band to obtain the preliminary result that the controller interacts with the drone on that band utilizing the spread spectrum technique.[23] This increases its resistance to interference, noise, and jamming. As a result, we could not jam its communication channel without advanced devices and tools. However, if the communication between the drone and the controller is disconnected, we can use the false direction guiding approach to force it to land on an attacker-specified area.

Other studies[3–5] have shown that all commercial aircraft and many general aviation aircraft continue to use old VOR/DME navigation equipment. Because VOR/DME signals are more powerful, they are less vulnerable to deception than GPS signals. Traditional VOR/DME devices can provide pilots with valuable cross-validation that can be compared with the GPS-generated position and speed. Compared with small UAVs, manned aircraft are usually equipped with a higher quality IMU, which means that the GPS navigation result can be cross-validated with the IMU more effectively. The simplest and easiest way to defend against GPS spoofing is probably to monitor the total received power near the GPS band of interest (for example, GPS L1). This can be done with an interference noise (J/N) sensor in the RF front end of the GPS receiver.

6. Conclusion

We presented three approaches to hijacking a flying drone by exploiting the vulnerabilities of civil GPS. Two approaches achieved their goals and one could not achieve its goal without advanced RF jamming techniques. We conducted all the hijacking experiments only with HackRF One, which is a cheap device and available for civilians, and open source projects without much programming.

There is still some room for improvement in hijacking a drone, and approaches are not limited to exploiting civilian GPS vulnerabilities. For example, in our experiments, a better hijacking attack can be made by additional control message jamming. The difficulty of attack depends on how the vendors/developers implemented their communication protocol. The DJI Phantom 3 Standard operates on the 5.8 GHz band with a bandwidth of almost 100 MHz using a spread spectrum technique, which makes the communication channel much more robust. There are various possible ways to jam this communication channel: jamming with an advanced device that covers such a wide bandwidth and deciphering its communication protocol and implementing an enhanced attacking algorithm on an SDR platform.

However, we mainly focus on the disclosure of civil GPS vulnerabilities in this paper. Hence, from the results of our experiments, we claim that current unencrypted civilian GPS signals endanger the safety and security of UAVs. We have the following suggestions on how to better secure drones' navigation systems. First, drones should be equipped with at least two types of GNSSs that transmit their signal on different bands. Furthermore, the Global Navigation Satellite System (GLONASS) encodes its signal using the frequency division multiple access (FDMA) technique, which has a higher resistance to interference. Receiving the satellite signal from different GNSSs makes spoofing by attackers more difficult. Second, an efficient double-check mechanism on location information via the IoT can solve the limited resource problem. The DJI drone uses the DJI GO app to monitor the hardware/firmware information of the co-connecting drone, showing it on a mobile device. We can leverage the IoT technique to gather all the hardware/firmware information and send it to external servers with the high performance of security monitoring and strong protection mechanisms. We believe that these two methods can greatly reduce the risk of UAV hijacking.

References

1 Amazon Prime Air: https://www.amazon.com/Amazon-Prime-Air/b?node=8037720011 (accessed March 2020).
2 D. Glade: July 2000 Center for Strategy and Technology Air War College. (CSTAWC, 2000) 0–38.
3 N. O. Tippenhauer, C. Popper, K. B. Rasmussen, and S. Capkun: CCS '11: Proc. 18th ACM Conf. Computer and Communications Security (ACM, 2011) 75–86.
4 T. Humphreys: Statement on the Vulnerability of Civil Unmanned Aerial Vehicles and Other Systems to Civil gps Spoofing (University of Texas at Austin, July 18, 2012).
5 K. C. Zeng, Y. Shu, S. Liu, Y. Dou, and Y. Yang: 2017 ACM in Proc. 18th Int. Workshop on Mobile Computing Systems and Applications (ACM, 2017) 85–90.
6 Software-defined GPS Signal Simulator: https://github.com/osqzss/gps-sdr-sim (accessed March 2020).
7 Global Positioning System: https://wikipedia.org/wiki/GlobalPositioningSystem (accessed March 2020).
8 J. S. Warner and R. G. Johnston: BCLC (2004) 19. http://the-eye.unblocksite.ch/public/Books/Electronic%20Archive/GPS-Spoofing-Countermeasures.pdf
9 Broadcast Ephemeris Data: https://cddis.nasa.gov/DataandDerivedProducts/GNSS/broadcastephemerisdata.html (accessed March 2020).
10 F. Kendoul: J. Field Rob. **29** (2012) 315. https://doi.org/10.1002/rob.20414
11 E. Vattapparamban, I. Guvenc, A. I. Yurekli, K. Akkaya, and S. Uluagac: Sept. 2016 IEEE Int. Wireless Communications and Mobile Computing Conf. (IEEE, 2016) 216–221.
12 S. M. Giray: June 2013 IEEE 6th Int. Conf. Recent Advances in Space Technologies (RAST) (IEEE, 2013) 795–800.
13 Software-definedradio: https://en.wikipedia.org/wiki/Software-definedradio (accessed March 2020).
14 HackRF, Open Source Hardware for Software-defined Radio: https://greatscottgadgets.com/hackrf/ (accessed March 2020).
15 HackRF-Clocking: https://github.com/mossmann/hackrf/wiki/Clocking (accessed December 2019).
16 HAK5-GPS Simulator: https://forums.hak5.org/topic/38290-gps-simulator/ (accessed December 2019).
17 Ubuntu Manpage HackRF Transfer: http://manpages.ubuntu.com/manpages/xenial/man1/hackrftransfer.1.html (accessed December 2019).
18 RINEX: https://en.wikipedia.org/wiki/RINEX (accessed December 2019).
19 FTP Server of GPS Broadcaste Phemeris Data: ftp://cddis.gsfc.nasa.gov/gnss/data/daily/ (accessed December 2019).
20 DEFCON24-Drones Hijacking-multi-dimensional Attack Vectors and Counter Measures: https://github.com/Aaron-Luo/DEFCON24 (accessed December 2019).
21 Dji Go App: http://www.dji.com/zh-tw/phantom-3-standard/app#sub-feature (accessed July 2017).
22 Dji Phantom3 Standard: https://www.dji.com/tw/phantom-3-standard (accessed July 2017).
23 Spread Spectrum: https://en.wikipedia.org/wiki/Spreadspectrum (accessed December 2019).