Security Vulnerabilities and Countermeasures of Unmanned Aerial Vehicle: An Experimental Study

***Abstract*—The abstract goes here.**

I. Introduction

The demand for UAV (Unmanned Aerial Vehicle) also known as drone technology is increasing very rapidly [**?**]. In recent year, drones are increasingly used not only in military applications, but also for civilian tasks, like delivery services, traffic surveillance, agriculture and farming, aerial surveys and other tasks that are too dangerous or remote for humans [**?**]. Soon drones can be used in disaster relief, expanding internet access and tasks too dangerous or remote for humans [**?**]. Recently, The German logistics company DHL and Amazon announced the launching of a new drone-based delivery service. An Australian textbook rental company, Zookal, has already started using drones to deliver books. Drones have become part of Internet-of-Things (IoT), because of increasing demand for use of drones in commercial areas, for example agricultural and surveillance drones collects vast amounts of visual data from the air and pass data to cloud using internet for processing.

In future drones will replace connected sensors at rest in IoT with one device deployable to different locations, capable of carrying flexible payloads and re-programmable in mission. Recently Brodcom introduced the WICED Sense Development Kit, an all-in-one IoT prototyping kit for drones. The Erle Robotics company offers Erle-Brain operating systems capable of connecting drones with IoT. In few years Internet connected drones will be widely used in commercial and civil tasks. However IoT Drones are directly accessible over the Internet, and cause serious threat to security. Recently, a drone-hijacking program called SkyJack has been engineered to autonomously seek out, hack, and wirelessly take over other drones within Wi-Fi distance and turn them into a zombie drone under attackers control. It is thus critical to address security requirements, such as authentication, authorization, non-repudiation, confidentiality and integrity, in order to IoT drones.

This work is organised into following sections. in Section III, we breifly describe the related works done in this field. We provided a detailed analysis of architecture and internal operations and governing system in Section **??**. In Section V, we explained our experiments like reverese engineering firmware, removing authentication from DJI SDK and attacks performed like GPS spoofing attack on P4P, wireless attacks on Bebop 2 drone and some propositions to counterattack those security threats in Section VII. In Section VI, we compared two drones in terms of user-friendliness and security, giving security primary importance.

II. Motivation

III. Related Works

Here we illustrate the survey of vulnerabilities found in Phantom and Bebop drones. Previous work on drone hacks include unprotected WiFi, access to config files, change settings in flight, GPS attacks. We have performed some known experiments and exposed some new vulnerabilities in our work reporting each for DJI Phantom 4 Pro and Parrot Bebop 2. The works in [1], several hackers in [2] in DEFCON demonstrated attacks mainly on DJI Phantom 3 Standard. We posit that most of them are not applicable for DJI Phantom 4 Pro. We found GPS spoofing as the most prominent attack on DJI drone, whereas it does not affect Bebop 2 drone.

A. WiFi insecurities

The Parrot drones allow multiple connection through WiFi, which allows attacker to tamper with the system and hack the drone. We verified different types of wireless attacks on the Bebop2 drone, but these attacks are not applicable to Phantom 4 Pro as there is no WiFi communication.

B. SkyJack

The Skyjack [3] developed by Samy Kamkar uses a Raspberry Pi and aircrack-ng [4] mounted on a WiFi-based drone, e.g. Phantom 2 Vision or Parrot drones, flies around and gets into the network of nearby drones. It disconnects the controller by changing SSID, immediately reconnects and is able to transmit commands through the malicious drone to the hijacked drone and takes control of other drone in flight using a customised script, hence forming zombie drones. DJI Phantom 4 Pro is not vulnerable to this attack whereas Bebop2 is.

C. GPS based attacks

GPS attacks are an ongoing area of research even with military GPS applications. Bebop 2 did not respond to this attack but GPS spoofing was successfully conducted on Phantom 4 Pro. [2] demonstrates the neccessary conditions to be met for successful spoofing and different capture and post-capture control techniques.

D. Maldrone

Maldrone [5] sets up a proxy serial port, intercepts flight commands from the controller and redirect actual serial port communication to fake ports, while forwarding the hijackers commands to the drone. While Bebop 2 will be affected, this attack is not applicable for Phantom 4 Pro.

IV. System Overview

In this section, we are presenting the detailed description of the internal architectures, internal operations and operating system of DJI Phantom 4 Pro and Parrot Bebop 2 drones.

A. DJI Phantom 4 Pro

Here we introduce a brief overview of the DJI Phantom 4 Pro drone system and its components.

(a) Controller with phone (b) Phantom 4 Pro

Fig. 1: DJI Phantom 4 Pro with controller

1) Remote Controller: The Phantom 4 Pro remote controller is a multi-function wireless communication device equipped with Lightbridge technology that integrates a dual frequency video downlink system and the aircraft remote control system. The remote controller features a number of camera control functions as well as gimbal control.

2) Mobile Device: The mobile device acts as an user-friendly interface to control the drone and displays the live video feeds. The system supports both IOS and Android device. The mobile app, i.e DJI GO 4 app serves as the controller for many advanced features like RTH(return-to-home), Waypoint mission and other intelligent flight modes like TapFly to fly in the designated position without using remote controller, ActiveTrack to mark/track a moving object on the mobile screen, e.g. tracking a car in a race, Draw for waypoint mission. The mobile device sends instructions and settings to the controller, and the controller relays these instructions to the drone via a 5.725 GHz - 5.825GHz radio signal.

3) Drone: The drone takes flight command and delivers video feed from the controller using radio signals of 2.400 GHz - 2.483 GHz and 5.725 GHz - 5.825 GHz ISM bands. 5 GHz video donwlink is recommended when ther is more interference. It also has the ability of 5-diectional obstacle sensing and a couple of other onboard sensors like barometer to stabilize itself.

DJI Phantom 4 Pro operates using Rockchip ARM-processor and Unix-based operating system-Busybox. It comes with a powerful remote controller having Lightbridge technology. Lightbridge is a one way stream of data with no two way handshaking, minimizing video latency while maximizing flight distance. Latencies with Lightbridge are more consistent at distance as a result, versus standard WiFi. Lightbride is much faster at re-establishing a connection when it reaches its distance limit. The GL300 main board is the most important hardware component supporting processor, Ambarella CPU, FPGA Altera Cyclon V, microSD slots, radio transceiver modules, etc.

The OFDM Receiver module is responsible for pairing with RC and communicating with it. The name comes from Orthogonal Frequency Division Multiplexing - which is a way of transferring broadband data. It implements the Lightbridge technology, and is also a middleman between the gimbal and flight controller. The control and video is transmitted over 2.4 GHz using digital transmission for the video along with the standard 'packet' style control link with the usual parity and identity confirmation checks over frequency hopping spread spectrum. Ambarella make video stream to Davinchi module via HDMI (or USB) and at the same time write video stream to microSD (H.264 or H.265) , then Davinchi send video to Encryption module Altera Cyclon V and then encrypted stream goes to RF transceiver (AD9364). GPS chip used is Ublox NEO which has the ability of J/N monitoring.

B. Parrot Bebop 2

Parrot Bebop 2 is an easy-to-fly user-friendly drone. It offers 25 minutes autonomous flight with digital stabilisation. It may come with a dedicated controller, SkyController to increase the flight range and smooth control using joysticks. But mobile device is sufficient for communication with the drone.

Fig. 2: Bebop 2 with SkyController

1) Drone: The Bebop drone communicates with the controller or/and mobile device via WiFi. The drone itself acts as a 802.11 WiFi hotspot for 2.400 GHz - 2.483 GHz and establishes a Bebop network once powered on. The WiFi modules allow a controller, either a mobile device or a dedicated controller like SkyController (increases telemetry range and provides physical joystick control) to connect to the network. Both the controller and the drone maintain their open access points(AP) active.

2) Mobile Device: The mobile device binds to the network, establishes connection with the drone via TCP. It sends flight commands and receives flight data and other sensor readings and sends video packets over UDP. In its current version, "Freeflight Pro" an official app by the Parrot Foundations using ARDrone SDK, it has several features like access to create autonomous flight plans.

Bebop 2 uses a P7 dual core CPU, quad-core GPU, and 8 GB of memory and operating system is a striped down version of Unix: Busybox. Bebop2 OS, ARDrone3 maintains the typical UNIX filesystem hierarchy as well as its open architecture. The Bebop memory is accessed through its WiFi connection and the use of communication protocols such as telnet and FTP. The onboard flight controller controls the drone by combining inputs from several sensors. These include:

a) GPS chip: After removing the Bebop plastic nose, the Furuno GN-87F GPS chip(or Ublox Neo 8M) is located on top of the camera mount block.

b) Others: The MS5607 barometer mounted onboard the Bebop has a known problem with light: when hit by strong light, the barometer undergoes a sudden shift. There are other sensors like accelerometers and magnetometers for assistance.

c) Vertical Camera: The vertical MT9V117 camera shoots a frame every 25 ms. Using these frames the software adjusts the Bebop orientation and XY position.

V. Attacks performed

A. DJI Phantom 4 Pro

1) Cracking DJI SDK: DJI provides mobile and on-board SDK which are available for downloads from [6]. This attack is based on removal of authentication between the DJI app and the DJI server. DJI offers an SDK to build custommised Android and iOS apps. This allows to modify the SDK by decompiling and patch portions of code. The first use of DJI Go or any customised app using DJI-SDK, requires one time registration and activation. We decompiled

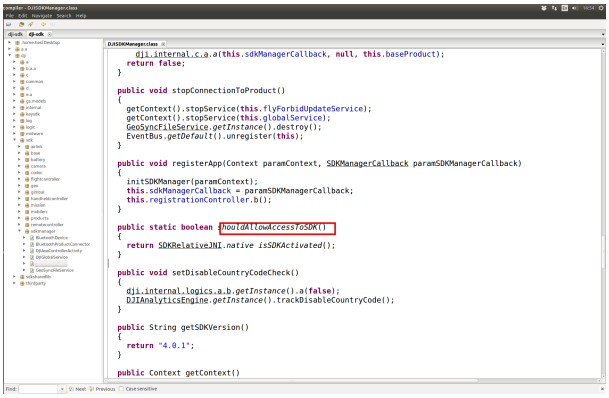


Fig. 3: Portion of decompiled SDKManager Class

the library jar file dji-sdk using [7] and edited the code using [8]. We found functions registerApp, isSDKActivated, shouldAllowAccessSDK, hasSDKRegistered mainly handling this authentication. startConnectionToProduct() which in turn calls shouldAllowAccessSDK(), checks whether the user will be given access to SDK, hasSDKRegistered() registered with a valid AppID(key) against a registered user, which is obtained after registration of an app. This key is pasted as AppID in the main Manifest.xml file of the customised app to activate DJI-SDK.

We patched two functions shouldAllowAccessSDK(), hasSDKRegistered() to return true always without any further permission checking, thus effectively removing authentication. So even a wrong key or null string as AppID would not throw an error preventing opening of the app. This authentication is crucial to prevent any malicious user taking control over a drone of an authentic user. We built some custom apps for controlling camera, taking photos, videos and a playback manager using the cracked SDK. The custom apps showed successfull authentication, worked perfectly fine.

2) Reverse engineering firmware: The binary of the DJI Phantom 4 Pro firmware is encrypted and digitally signed. Though we were able to decrypt the binary, it was not possible to reflash with the new binary. The reason behind it probably being the binary not completely decrypted. We ran a customised parser and obtained two modules. Each of the module was first unpacked using rkflash [9]: a flashing tool for Rockchip based systems, and obtained standard images. In the Figure **??**, the image *embedded-update.img* was further extracted into standard image modules: *boot.img*, *recovery.img*, *system.img*. They were further unpacked into standard Android partitions using different Android image flash tools.

3) GPS Spoofing: DJI Phantom drones are GPS based drones; receiving incoming signals from GPS satellites, signals notifying the drones presence, and a two-way link between the ground station controller and the drone. GPS enables a drones navigation, and due to no encryption of the civilian GPS signals they can be easily spoofed. The basic idea in GPS spoofing is transmitting fake GPS coordinates to the flight controller of the drone. This will hijack the drone and subsequently it will be in complete control of the attacker. To spoof a GPS receiver, a transmitter is used to transmit false GPS signals forcing the victim to synchronize with the attackers signals, including spoofed information regarding the ephemeris and almanac data. If the GPS receiver switches GPS fix from legitimate GPS signals to false GPS signals, drone deters its path and sends the drone in a direction specified by the attacker. A successful attack is conducted when attacker is very close to the drone, or by using a directional antenna of proper frequency aiming the drone.

Though DJI improved the navigation by inertial measurement unit(IMU) instead of relying only on GPS. Still we managed to spoof the GPS receiver using LabSat3 GPS simulator [10]. DJI Phantom 4 Pro comes with Ublox Neo 8M GPS receiver which has the ability to detect jamming with the help of J/N monitoring. J/N monitors triggers an alarm if the recieved power is more than the trigerring threshold, preventing jam-and-then-spoof kind of attack. We performed the spoofing attack inside a screened environment in the presence of weak real GPS signals. It was difficult to conduct the experiment

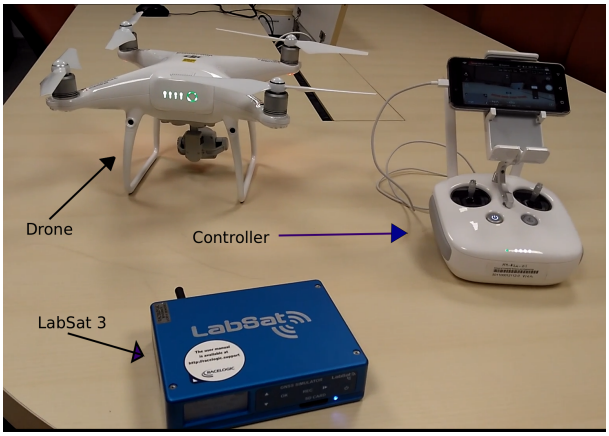


Fig. 4: GPS Spoofing Setup

outdoors due to the presence of strong legitimate GPS signals.

Experiment Procedure: Here we perform record and replay attack only. First real GPS signals are recorded using theactive antenna RLACS198 provided with the labsat kit. This antenna does not work as a transmitter. For receiving signals, antenna is connected to SMA RF IN port. Fake GPS signals can also be generated using SatGen software provided with LabSat. There is no tansmitter antenna provided with the kit. To replay, it needs a 1575 MHz (GPS L1 frequency) antenna for proper working and connected to RF OUT port.

The location of the drone was successfully spoofed by LabSat.

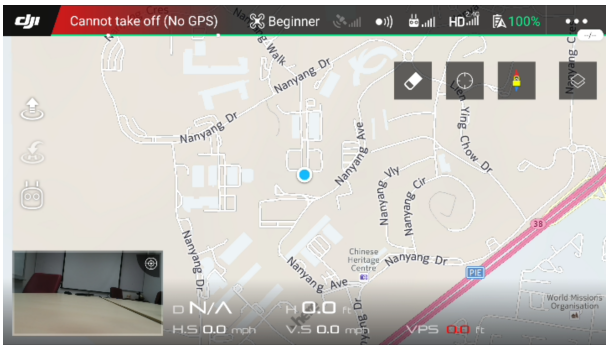


Fig. 5: Initial location: NTU, Singapore

The drone was placed inside HESL lab at NTU. Though DJI Phantom 4 Pro has ublox GPS module with J/N monitoring, it fails to detect spoofing in our case, as the transimitted power was very less. The location, altitude, speed is shown by the DJI Go 4 app. Several kind of hacks are performed: force landing a drone spoofing the location to be a NFZ, flying a drone in a NFZ, height hack. In the figure 5, the drone is seen to be located at Nanyang Technological University, the spoofed location shows the drone to be in Changi South Avenue near SUTD in figure 6, which is a no-fly-zone. If it had been flying with real GPS, and we transmit fake GPS, and these are more stronger in quality than real ones and the spoofed location is a NFZ, the drone would perform an emergency landing.

Similarly the reverse experiment could be performed where

Fig. 6: No Fly Zone near SUTD, Changi Airport

the drone was located in a NFZ preventing the drone to take-off and if location is successfully spoofed to a non-restricted place, drone would take off.

Also the movement of the drone could be faked by replaying any motion curve or pre-recorded data as in figure 7 below.

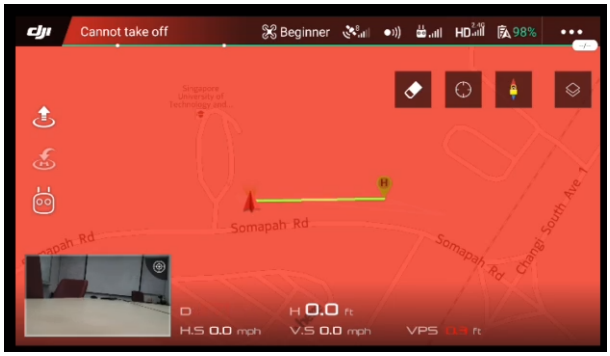


Fig. 7: Drone Movement

B. Parrot Bebop 2

Most of the attacks in Parrot Bebop 2 being tested are possible to the open WiFi. Even the Parrot manufacturers have introduced the feature of adding WPA/WPA2 password to the Bebop WiFi which obvioulsy decreases range and speed. Nonetheless, it remains vulnerable to WiFi attacks. Weak password can be easily cracked by tools like [4].

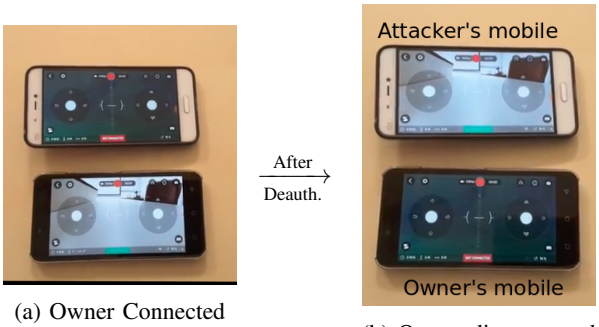
Network Mapping: The nmap utility shows the ports used by the drone and the different services using it. By nmap scan of the drone's IP, we got 3 open ports running services ftp, telnet, tor-socks.

1) Open WiFi: Bebop 2 drone uses an IEEE WiFi radio type of 802.11ac and the IP address of the drone is 192.168.42.1. An unencrypted open WiFi allows any one to connect and hijack the drone. Since Bebop2 allows for multiple clients to connect to the network, the problem is more critical i.e. more than one user can control the drone. There is no way of validating the original owner.

2) Deauthenticating owner: We performed the deauthentication attack on the drone using [4]. This attack consists of capturing packets of the Bebop network without even connecting to it, diconnecting the authenticated owner. Initially the

wireless network is scanned in monitor mode using airmon-ng. After the Bebop network is found, it snoops into the network using airodump-ng capturing all the packets from that network only. When the list of clients are available, deauthentication attack is performed by executing aireplay-ng, which sends de-authentication packets continuously to the clients, disconnecting them until the program executes. In the meantime, the hacker gets connected and can hijack the drone being the only client connected.

In the Figure **??**, deauthentication packets are send continuously from the attacker's laptop to the connected client whose MAC address is 48:88:CA:62:9B:A3 where the drone's BSSID is A0:14:3D:C1:F4:2B. The entire network can be jammed by sending disauthentication packets continuously, preventinf anyone to connect to the network. The aireplay-ng sends 128 deauth packets: 64 packets to BSSID and 64 packets to the client. In each last column of figure **??**, first number denotes the number of ACKs received from the client and second denotes that received from the drone. In the above Figure 8, the



(a) Owner Connected

(b) Owner disconnected

Fig. 8: Deauthenticating owner's mobile

below mobile belonging to the owner remains disconnected as long as aireplay continues to send directed deauth packets. The attacker takes this advantage to connect his own mobile(above) and can hijack the drone.

3) Open Telnet: Once connected to the Bebop 2 WiFi, the user can telnet to it. The drone's IP address is 192.168.42.1. Since multiple user can connect to the drone WiFi, just attempting to connect to the network, generates multiple requests in a short span of time, which may cause **DOS** attack, results in interrupting original commands. Even by telnetting to it, the user get root access to the entire file system. By telnetting into the drone, we were able to kill the main process *dragon-prog*, which would stop motor arms immediately making the drone fall on the ground. *dragon-prog* is the main Bebop process governing the whole system of flight control. We also found a list of other interesting processes running by *ps* command of Unix. We also found important shell scripts */usr/bin/ardrone3_shutdown.sh*, */usr/bin/DragonStarter.sh*

We observed executing *ardrone3_shutdown.sh* also stopped the main process and the drone shuts down immediately.

4) Root access to file system: Since the release of AR.Drone 3.0, most of the files are read-only, except the /data directory which is the FTP media directory. Though the files are by default read-only, the user can override it by remounting the entire file system by

```
# mount -o remount, rw /
```

5) Open FTP port: Without any authentication for FTP, attackers may connect to the WiFi and login to the FTP to steal personal information including media, flight data, etc. Due to factory settings, FTP access does not allow to navigate the whole filesystem, but only the ftp directory */data/ftp/internal_000*, which is where images and videos are recorded, together with other useful files like pud files. The limited FTP access can be taken over by editing the file */etc/inetd.conf*. After the following lines,

```
21 stream tcp nowait root ftpd ftpd wSS /data/ftp
51 stream tcp nowait root ftpd ftpd wSS /update
61 stream tcp nowait root ftpd ftpd wSS
```

we need to add the following line to get access to entire file system:

71 stream tcp nowait root ftpd ftpd SS /

6) Snooping into the WiFi and packet capture: We used Wireshark to capture packets during the connection establishment and throughout the entire flight. The flight commands and video are passed as UDP packets while the initial connection setup is done by TCP.

7) Additional vulnerabilties:

a) Reversing firmware: The Bebop 2 firmware is completely reversible and can be easily modified and the new binary can be reflashed. Reversing firmware gives insight into the drone workflow and file system, which makes easier for the attacker to exploit vulnerabilities and target specific location.

b) Modifying files: The entire file system can be accessed using telnet after being connected to the Bebop's WiFi. Wrong modification of configuration files/scripts like dragon.conf to crash the software and hamper normal operation of the drone. Modifying /etc/passwd file may brick the drone as changing some passwords may cause the owner not be able to telnet into it, or connect to the drone.

VI. Comparison of two drones

From the different experiments conducted and keen observations made after analysis the security aspects of both the Phantom 4 Pro and Bebop 2 drones, the comparison can be made on the basis of easiness to fly, image quality, processing speed, smoothness, security, etc. But the most significant of all is the security aspect of drones.

It can be concluded that though Bebop 2 is more user-friendly, Phantom 4 Pro is more secure than Bebop 2. Phantom 4 Pro is only vulnerable to GPS spoofing and interception/jamming of radio signals using SDR. Bebop 2 is prone to different wireless attacks and can be easily hijacked. While SDR equipments and/or GPS spoofer are difficult to acquire in terms of availability and cost, several wireless tools and network scanner, analysers are readily available online free of cost. We also noticed that DJI tried to improve the security by introducing communication and video feed transmission via radio signals. DJI claims to have on-board FPGA video encryption algorithm

which can be easily verified scanning transmitted radio signals using SDR or tapping the output of FPGA before it goes to the transceiver. DJI also encrypts the firmware partially and the binary is digitally signed, still it can be decrypted to some extent.

VII. Proposed countermeasures

Below we propose some methods/actions that can be implemented to increase the security of drone. Drones shall remain vulnerable to GPS spoofing unless receiver also have the capability to detect spoofing like SAASM used by military-thus need for developing anti-spoofing and anti-jamming receivers. While there has been a lot of promising work and proposed methods in detection and avoidance of civilian GPS anti-spoofing and anti-jamming in the literature similar to that in [11]. Those methods can be broadly classified into cryptographic: spread spectrum, dual receiver correlation and non-cryptographic methods: antenna array, requires external hardware. The above methoda are either difficult to implement or requires costly hardware. So we would like to propose simpler software based techniques for spoof detection:

Checking latency: The motion speed can be validated for the change in location in a short instant, for e.g. one cannot travel and change its location from new York to Beijing in few seconds. The changes of coordinates can be recorded and validated given the time taken to change the coordinates.

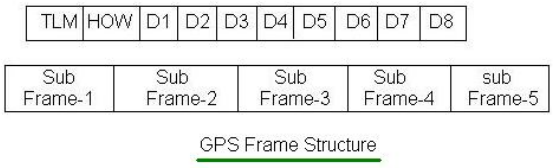


Fig. 9: GPS subframes

Checking GPS subframe data: The GPS frame starts with telemetry word used as preamble, which provides details of the satellite. Handover word(HOW) provides the GPS system time. HOW is followed by eight data words with parity bits. Subframe-1 carry information to correct the satellite clock. Subframe-2 and subframe-3 provides ephemeris data for the satellites. Subframe-4 and sunframe-5 send the Almanac data. GPS subframe data can be validated easily by checking each of the subframes. Fake GPS data is observed to have wrong subframe data, thus able to detect spoofing. Thus an onboard Raspberry Pi on the drone validating time-motion or GPS subframe data can easily detect gps spoofing.

Apart from GPS anti-spoofing and anti-jamming, we would like to recommend some modifications which can improve the security of drone.

- Using encryption/packer to protect library files
- Using obfuscator to prevent decompiling, reverse-engineering files
- Improve SDK authentication, now being done only between the app and server, drone must also be included in the one-time authentication
- Encrypt the entire firmware binary, the encryption key must be stored in the hardware

On the contrary, many of the attacks described above for Bebop 2 are possible only due to open WiFi and open ports. Additionally FTP need not be activated inflight to prevent stealing of flight data and media. This is enough to secure personal data if after flight this data can be backed up offline. For Bebop 2, generally WiFi-based UAVs, we propose the following countermeasures:

a) Adding WPA security: With the release of ARDrone3, Parrot developers introduced the feature of adding WPA/WPA2 security to the WiFi using Freeflight Pro app. Nonetheless with powerful WiFi password cracking tools like aircrack-ng, it is possible to crack the password. Also augmenting security causes decrease in range and transmission speed.

b) Adding telnet password: Bebop 2 should not allow root access to the file system once connected and telnetted. This can be managed by adding a telnet password modifying */etc/passwd* file.

c) MAC-filtering and Hidden SSID: These can be achieved through bcmwl program installed in the drone. bcmwl controls all the wireless wi-fi connectivity of the drone. Any changes made can be temporary or permanent. In order to reversibly hide the wifi SSID, open a telnet session and enter the following command:

```
# bcmwl closed 1
```

This makes it more difficult for others to connect to the drone, since the wifi network will not appear in the list of available networks.

To activate MAC address filtering,

```
# bcmwl mac MA:CA:DD:RE:SS:01 MA:CA:DD:
RE:SS:02 ...
```

where all the possible MAC addresses are given by the owner followed by:

```
# bcmwl macmode 2
```

which sets MAC filtering to only accept connections from the previous list of devices. To permanently introduce these changes, we need to modify */sbin/broadcom_setup.sh* file.

VIII. Future Work and Conclusion

We had to leave some portion as future work due to unavailability of proper hardware. We did not yet scan the radio signals through which the P4P communicates. It can be scanned and analysed using SDR equipments like HackRF, BladeRF. As further work, the transmission of video and control commands using packet style can be studied which DJI claims that it uses frequency hopping spread spectrum(FHSS). Though DJI manufacturers claim that the live video feed transmitted is encrypted using on-board FPGA encryption algorithm, it is yet to be verified. As further work for the Bebop 2 drone, we look forward to build Arducopter-an autonomous flight controller on top of it.

In conclusion, the P4P is one of the most secure and robust

drones in the commercial market. Though DJI has developed P4P such that it is less vulnerable than its preedecessors, still it needs further work and a security analysis is worthwhile. Whereas Parrot manufacturers did not employ any security mesure for Bebop drones, Bebop 2 is too risky for any personal or commercial use. We believe the recommendations proposed as countermeasures would improve the security of Bebop drone and make it suitable for nominal use.

Future smart cities will be swarming with drones for different commercial and personal purposes in the hope that it will improve the lives of human being. With increasing reliability and dependency on technology, vulnerability becomes prominent and security becomes a chief concern. Drone manufacturers must employ techniques to improve the drone security, make it robust to attacks. Even if the drone can not be made entirely fool-proof, it must be resistant to easy wireless hacks and signal jamming or interception.

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] Fernard Trujano, Benjamin Chan, Greg Beams, and Reece Rivera. Security analysis of dji phantom 3 standard. 2016.

[2] Andrew J. Kerns, Daniel P. Shepard, Jahshan A. Bhatti, and Todd E. Humphreys. Unmanned aircraft capture and control via gps spoofing. *Journal of Field Robotics*, 31(4):617–636, 2014.

[3] Samy Kamkar. Skyjack: autonomous drone hacking. *Available: http://samy.pl/skyjack*, 2013.

[4] Aircrack-ng: tools to access wifi network security. *Available: http://www.aircrack-ng.org*.

[5] Maldrone: Backdoor for drones. *Available: http://garage4hackers.com/entry.php?b=3105*, 2015.

[6] Dji sdk for developers. *Available: https://developer.dji.com*.

[7] Jd-gui: standalone java decompiler. *Available: http://jd.benow.ca*.

[8] Java bytecode editor. *Available: http://set.ee/jbe*.

[9] rkflashtool: tools for flashing rockchip devices. *Available: https://github.com/neo-technologies/rkflashtool*.

[10] Labsat 3: A multi-constellation global navigation satellite simulator. *Available: https://www.labsat.co.uk/index.php/en/products/labsat-3*.

[11] Kyle Wesson, Daniel Shepard, and Todd Humphreys. Straight talk on anti-spoofing. *GPS World*, 2012.