University of Derby
Department of Electronics, Computing and Mathematics

A project completed as part of the requirements for
BSc (Hons) Computer Forensic Investigation

entitled
A Forensic Analysis of the Parrot AR Drone 2.0

by
Kheia Grace Maune

May 2018
Abstract

Unmanned Aerial Vehicles (UAVs) have become increasingly popular amongst consumers and businesses. There is no doubt that they have their beneficial uses. However, there has been an increase in UAV’s being misused. This, in turn, has led to an increase in reports of UAV-related crime and offences. With the government struggling to legislate quick enough to keep up with these incidents, there is a need for law enforcement to be able to forensically acquire, examine and analyse a UAV and its peripheral components.

The aim of this paper is to perform an in-depth forensic analysis of the Parrot AR Drone 2.0 GPS Edition and its peripheral components, including the Flight Recorder and a Samsung Galaxy S4 Mini as the flight controller. Previous research and literature was reviewed, and various scenarios were created to generate flight data including GPS, photo and video data. This project found that it is possible to retrieve GPS data and media files containing EXIF data from both the Parrot AR Drone 2.0 and the flight controller. Only media files containing EXIF data were retrieved from the Flight Recorder. Despite these findings, problems still exist in identifying the definitive owner of a UAV. It is hoped that the findings presented in this paper can be used to contribute to future research and investigations.
Acknowledgements

Thank you to my family and friends. Your constant patience, support, encouragement and motivation has been invaluable. To my family, I hope I have made you proud.

Thank you to the teaching staff at University of Derby who have helped me to further my knowledge and skills. Thank you to Virginia Franqueira for your advice, supervision and guidance throughout this project.

Finally, thank you to my classmates. Each one of you have made my University experience a memorable and enjoyable one. I wish each of you the best for the future.
# Table of Contents

1. Introduction ...................................................................................................................... 12

1.1. Project Rationale ........................................................................................................ 12

1.2. Project Aim and Objectives ....................................................................................... 14

2. Literature Review ............................................................................................................. 15

2.1. Digital Forensic Investigation .................................................................................... 16

2.1.1. Forensic Soundness ............................................................................................ 17

2.1.1.1. Hashing Algorithm ..................................................................................... 17

2.1.2. Forensic Methodology ........................................................................................ 17

2.1.2.1. McKemmish ............................................................................................... 18

2.1.2.2. An Abstract Digital Forensics Model ......................................................... 18

2.2. Unmanned Aerial Vehicles ........................................................................................ 20

2.2.1. Classification of UAV Systems .......................................................................... 20

2.2.2. Basics of a UAV Multicopter ............................................................................. 21

2.3. Parrot AR Drone 2.0 GPS Edition ........................................................................... 23

2.3.1. Technical Specifications ..................................................................................... 23

2.3.2. Parrot Mobile Application .................................................................................. 24

2.3.3. Flight Recorder ................................................................................................... 25

2.4. UAV Forensics .......................................................................................................... 26

2.4.1. UAV Methodology ............................................................................................. 26

2.4.2. DJI Forensics ...................................................................................................... 26

2.4.3. Parrot Forensics .................................................................................................. 28

2.5. UAV Security ............................................................................................................ 29
2.5.1. Parrot AR Drone 2.0 Vulnerabilities ................................................................. 29

2.6. UAV Legislation ........................................................................................................ 31

2.6.1. Small Unmanned Aircraft .............................................................................. 32

2.6.2. Small Unmanned Surveillance Aircraft ......................................................... 33

2.6.3. Future Legislation ......................................................................................... 33

2.7. Mobile Controller ............................................................................................... 34

2.8. Mobile Forensics .................................................................................................. 35

2.8.1. Mobile Forensic Challenges ........................................................................ 36

2.8.2. Mobile Device Tool Classification System ................................................... 37

2.9. Android ............................................................................................................... 38

2.9.1. Android System Architecture ...................................................................... 39

2.9.2. Android Boot Process .................................................................................. 41

2.9.3. Android File System .................................................................................... 42

2.9.4. Android Forensic Data Acquisition .............................................................. 43

2.10. Linux ................................................................................................................. 45

2.10.1. Linux System ............................................................................................ 45

2.10.2. Linux File System ....................................................................................... 47

2.11. Conclusion .......................................................................................................... 47

3. Methodology ........................................................................................................... 49

3.1. Main Components .............................................................................................. 50

3.2. Tools .................................................................................................................. 50

3.3. Applied Methodology ....................................................................................... 51
3.3.1. Parrot AR Drone 2.0 GPS Edition ................................................................. 51
3.3.2. Samsung Galaxy S4 Mini ............................................................ 51
3.4. Data Generation .................................................................................. 52
3.5. Data Analysis ...................................................................................... 53
  3.5.1. Parrot AR Drone 2.0 GPS Edition ...................................................... 54
    3.5.1.1. Tools/Software ................................................................. 54
    3.5.1.2. Key Areas ........................................................................ 54
  3.5.2. Samsung Galaxy S4 Mini Flight Controller ......................................... 54
    3.5.2.1. Tools/Software ................................................................. 54
    3.5.2.2. Key Areas ........................................................................ 54
  3.5.3. Flight Recorder ............................................................................. 55
    3.5.3.1. Tools/Software ................................................................. 55
    3.5.3.2. Key Areas ........................................................................ 55
  3.5.4. Micro SD Card ............................................................................. 55
    3.5.4.1. Tools/Software ................................................................. 55
    3.5.4.2. Key Areas ........................................................................ 55
  3.6. Ethics ............................................................................................... 56
  3.7. Limitations ....................................................................................... 56
  3.8. Conclusions ..................................................................................... 56
4. Results and Findings ............................................................................... 57
  4.1. Limitations ....................................................................................... 57
  4.2. Data Generation Revisited ............................................................... 58
  4.3. Parrot AR Drone 2.0 GPS Edition .................................................... 60
    4.3.1. Unsorted Block Image File System ............................................. 60
4.3.2. Flight Data .......................................................................................................... 61
4.3.3. Media .................................................................................................................. 65

4.4. Samsung Galaxy S4 Mini Flight Controller ............................................................. 66
   4.4.1. Flight Data .......................................................................................................... 66
   4.4.2. Media .................................................................................................................. 67
   4.4.3. Other Relevant Data ........................................................................................... 68

4.5. Flight Recorder .......................................................................................................... 69
   4.5.1. Flight Data .......................................................................................................... 69
   4.5.2. Media .................................................................................................................. 69

4.6. Counter Forensic Methods ......................................................................................... 69
   4.6.1. Parrot AR Drone 2.0 – Time and Date Manipulation ....................................... 69
   4.6.2. Samsung Galaxy S4 Mini – Time and Date Manipulation ................................ 70
   4.6.3. Flight Recorder – Time and Date Manipulation .............................................. 71
   4.6.4. Parrot AR Drone 2.0 – GPS Manipulation ....................................................... 72
   4.6.5. Samsung Galaxy S4 Mini – GPS Manipulation ............................................... 75

4.7. Conclusion ................................................................................................................. 75

5. Discussion and Analysis ................................................................................................... 76
   5.1. Key Findings .............................................................................................................. 76
   5.2. Relevance ................................................................................................................... 78
       5.2.1. Compared to Previous Literature ........................................................................ 78
       5.2.2. For Law Enforcement ....................................................................................... 79
   5.3. Correlation Between Main Components ................................................................... 80
Table of Figures

Figure 1. Communication Between UAV and Mobile Flight Controller. .................................. 16
Figure 2. Forensic Methodology Proposed by McKemmish (1999). ..................................... 18
Figure 3. An Abstract Digital Forensics Model (Reith, Carr and Gunsch, 2002). ............... 19
Figure 4. Components of a Quadcopter UAV System (Parrot, n.d.). .................................. 22
Figure 5. AR.FreeFlight.2.4.15 Home User Interface (Google Play, n.d.). .......................... 24
Figure 6. AR.FreeFlight 2.4.15. Piloting User Interface (Google Play, n.d.). ....................... 25
Figure 7. Global Mobile OS Market Share 2009-2017 (Statista, 2017). ............................... 35
Figure 8. Mobile Device Tool Classification System (Brothers et al., 2014). ......................... 37
Figure 9. Worldwide Smart OS Market Share 2014-2021 (Statista, 2017). .......................... 39
Figure 10. Android System Architecture (Android, 2018). .................................................. 40
Figure 11. Linux System Architecture (Ward, 2015). ........................................................... 46
Figure 12. Userbox File for Standard Flight Converted to .GPX .......................................... 63
Figure 13. Standard Flight Coordinates in Google Earth. .................................................... 64
Figure 14. Userbox File Without Flight Recorder ................................................................. 64
Figure 15. Coordinates from /data/custom.configs/sessions/config.XXXXXXXX file in Google Earth. ..................................................................................................................... 65
Figure 16. Userbox File Showing Manipulated Time and Date ............................................. 70
Figure 17. EXIF Data of Video File showing Correct Time and Date ................................... 71
Figure 18. GPS Manipulated Coordinates in Toronto, Canada ............................................ 72
Figure 19. GPS Data stored in Userbox File for GPS Manipulated Flight. ............................ 73
Figure 20. GPS Data stored in Userbox File as Plotted in Google Earth ............................. 73
Figure 21. Manipulated GPS Data Stored in "config.XXXXXXXX.ini" as Plotted in Google Earth. .............................................................. 74

Figure 22. Relationship between the Main Components to provide GPS Data. .................87

Figure 23. Relationship between the Main Components for Time and Date. .....................88

Figure 24. Relationship between the Main Components for storing Media files...............88
Table of Tables

Table 1: Summary of findings identified on Parrot AR Drone 2.0. ........................................ 77

Table 2: Summary of findings identified on Samsung Galaxy S4 Mini. ................................. 78

Table 3: Summary of findings identified on Flight Recorder. .................................................. 78
1. Introduction

1.1. Project Rationale

An Unmanned Aerial Vehicle (UAV), often referred to as a ‘Drone’, is an aircraft which flies without an onboard pilot or passengers. A UAV is typically controlled remotely by a pilot on the ground using a controller. However, UAV’s are becoming increasingly autonomous and less dependent on human intervention (Miller, 2015).

In 2010, Parrot unveiled the first generation in their AR Drone series at the International Consumer Electronics Show (CES) in Las Vegas. The AR Drone gained a lot of press and attention since it could be controlled using an iPhone. In 2013, Da-Jiang Innovations (DJI) unveiled the first generation in their Phantom series. The popularity of which would see DJI go on to hold a majority of the market (Juniper, 2016). Since then, UAV’s have become increasingly popular amongst the commercial and consumer market.

The UAV market is estimated to reach $100 billion by 2020. With growing demand, the consumer market is estimated to make up $17 billion of this. It is also estimated that there will be around 7.8 million consumer UAV shipments and $3.3 billion revenue in 2020, versus the 450,000 shipments and $700 million seen in 2014 (Goldman Sachs, n.d.).

Since 2013 UAV’s have advanced significantly, providing better features at a relatively affordable price. Some UAV’s offer features such as thermal imaging, pollutant sensors, high definition camera, and more. There is a direct correlation between the advanced features of a UAV and the price an individual will pay. UAV’s have their uses in many instances, those of which include aerial photography and in agriculture for monitoring crops and livestock conditions (Miller, 2015). They also have their uses in low-cost, high-resolution aerial mapping. UAV’s have been suggested as an alternative for gathering information, 3D reconstruction and mapping of a disaster site (Spranger et al., 2016). Once deployed, a UAV can quickly and safely reach an area that an individual cannot. Similar use cases include supplying essential items in a disaster, geographic mapping of inaccessible terrain or locations and performing building safety inspections. UAV’s also have their uses in the commercial sector. In 2013, Amazon announced its plans for ‘Prime Air’. A UAV delivery service promising a 30-minute or less delivery time. Due to the constant changes in regulation, Prime Air is still in its development stage (Miller, 2015).

It is expected that the next generation of UAV’s will widen the gap between manned and unmanned flight. Adding greater stealth, sensory, payload, range, autonomy, and communication capabilities (Goldman Sachs, n.d.).
On the one hand, UAV’s have clearly had a positive influence. However, the threat which involve them being misused is growing. There has been an increase in UAV-related crime and offences. Between 2014 and 2015, reports of UAV-related incidents in the United Kingdom (UK) rose by 352%. Most of these reports involved a concern for public safety. Burglary and spying were also among those reported (Yeung, 2016). Other UAV-related incidents which have cropped up in the media involve flying in restricted airspace, illegal streaming of sports, and contraband being smuggled across borders or into prisons. Recently, a UAV was spotted flying in restricted airspace around Gatwick Airport. This led to the closure of the runway and forced five flights to be diverted (BBC, 2017). In 2017, ten people were sentenced for flying an estimated £1 million worth of prohibited items into prisons around the UK (BBC, 2017). An individual was also recently sentenced to 12 years in prison for attempting to smuggle 13 pounds of methamphetamine worth $46,000 across the US border (Davis, 2018).

Governments are struggling to legislate quick enough to keep up with increase in UAV-related incidents. For this reason, there is a need for digital investigators, who understand the relevance of a UAV in an investigation and can complete a forensic acquisition and analysis of these devices. However, the complexity of the UAV ecosystem makes this a challenging task. First, the UAV itself can vary in make, model, operating system (OS), components, features, and more, each of which present individual challenges. Second, there are the other peripheral components to take into consideration. This includes the flight controller and memory card(s). The flight controller may be a mobile phone device or a handheld control. A mobile phone device acting as a flight controller presents further challenges for a digital investigator. These are discussed in Section 2.8.1.

Finally, questions are asked such as “How can a criminal pilot be identified if only the UAV is found at the crime scene? Or if only fragments from a wreckage are found? Or when only the flight controller or UAV is found?” (Marks, 2017). James Mackler, an attorney specialising in UAV litigation, states “UAV forensics are becoming increasingly important as more UAV’s take to the air” (Marks, 2017).

This paper presents an in-depth forensic analysis of the Parrot AR Drone 2.0 GPS Edition and its components. This includes the Samsung Galaxy S4 Mini as the flight controller, and the Flight Recorder. The remainder of this paper will be organised as follows: Chapter 2 presents a review of current literature related to the components of this project including UAV’s, mobile phone and Android devices, and Linux systems. Chapter 3 presents the methodology and design
for the investigation component of this project. Chapter 4 presents the results and findings identified during examination. Chapter 5 presents a discussion and analysis of the findings identified in the previous chapter. Finally, Chapter 6 presents the conclusion of this project and recommendations for future work. Appendix A presents the 12 phases of the forensic UAV framework proposed by Jain et al. (2017). Appendix B presents a complete log of actions taken during the investigation aspect of this project.

1.2. Project Aim and Objectives

The aim of this study is to perform an in-depth forensic analysis of the Parrot AR Drone 2.0 GPS Edition and its peripheral components including the flight controller and Flight Recorder.

The aim of this study will be achieved by completing the following objectives:

- To review relevant literature related to UAVs and mobile forensics.
- To design and implement a suitable practical experiment to gather data.
- To analyse the data acquired during testing and identify the areas of importance.
- To discuss the findings of the project and how this could be considered important in the future of forensic investigations.
2. Literature Review

This chapter aims to complete the first objective of this study by presenting a review of current research and literature related to this study.

Section 2.1 presents a general discussion of Digital Forensic Investigation. This section introduces forensic soundness and why it is important. Section 2.2 presents a general discussion of UAVs, how they may be classified, and the basic features of a quadcopter UAV. The Parrot AR Drone 2.0 is introduced in Section 2.3. This section presents the technical specifications of this UAV and the features available, including the Flight Recorder. A discussion of current research and literature related to UAV forensics is presented in Section 2.4. Literature related to the security vulnerabilities of the Parrot AR Drone 2.0 is presented in Section 2.5. Current and future UAV legislation is presented in Section 2.6. As the second component to this investigation, an introduction to Mobile Phone devices is presented in Section 2.7. A discussion of Mobile Forensics is presented in Section 2.8. The challenges related to Mobile Forensics are also highlighted in this section. The Android OS is presented in Section 2.9. This section includes an in-depth look into the underlying Android system architecture and how a forensic acquisition of an Android device can be achieved. Finally, a brief discussion of Linux is presented in Section 2.10.

Figure 1 presents a basic example of the communication between the flight controller and UAV. The user must ensure that the correct flight application is installed on the flight controller in order to use the features of the UAV. Real time piloting is achieved as signals and commands are issued over Wi-Fi by the user to the UAV via the flight application. The (Global Positioning System (GPS) receiver embedded in the UAV receives coordinates from satellites, which provide positioning and flight data. Some UAV’s offer a feature which allow a user to define a pre-set flight plan on the flight controller. The coordinates of this flight plan are received by the UAV over Wi-Fi. The UAV will execute the pre-set flight plan based on the signals and coordinates received from the satellites.
2. The UAV will have its own WiFi receiver embedded. This will emit a WiFi network. Person A can then connect the mobile flight recorder to the UAV via WiFi.

1. Person A must first ensure that the mobile flight controller has the relevant application installed on it. Without this, Person A cannot use the features of the UAV.

3. With the flight application installed and the mobile flight controller connected to the UAV, Person A can issue commands and use the features of the UAV.

4. Global Positioning Signals (GPS) are received from satellites. GPS signals provide coordinates to the UAV. This can be used to load maps on the mobile flight controller so the UAV can execute a flight plan.

Figure 1. Communication Between UAV and Mobile Flight Controller.

2.1. Digital Forensic Investigation

Digital forensic investigation is defined as “the process of identifying, preserving, analysing and presenting digital evidence in a manner that is legally acceptable” (McKemmish, 1999). Digital forensic investigation is the umbrella term used to encompass all digital devices capable of storage. Similar areas include computer forensic investigation and mobile forensic investigation. These terms are used to define specific branches of digital forensic investigation. It is the role of the digital investigator to examine, identify and analyse the data on a digital device to aid an investigation. According to Locard’s Exchange Principle when two objects come into contact with one another, a trace of evidence is always left behind. The same principle can be applied to digital forensic investigation where traces of digital data can be recovered on devices, even when a criminal believes that the data has been deleted (Arnes, 2018).
2.1.1. Forensic Soundness

Electronic data is volatile. There is an ongoing battle to acquire an exact copy of electronic data without altering anything in the process. Data must be acquired, analysed, and reported in a way that is admissible in a Court of Law. McKemmish (1999) proposed four key principles to guarantee forensic soundness.

The first is “Minimal Handling of the Original”. An investigator should keep the application of computer process to original data to a minimum in order to minimise the likelihood of data being changed. Ideally, digital data should be duplicated. This should be done in such a manner, using forensic tools, that the duplicate is an exact copy of the original.

The second is to “Account for Any Change”. During an examination, it may be necessary for either the original or duplicate to be altered. If this is the case, an investigator must understand the nature of the change, and why this has happened. This should be fully documented.

The third is to “Comply with the Rules of Evidence”. One of the fundamentals is to ensure that the application of tools and techniques do not lessen the admissibility of the final product. The type of tools and techniques used, and the way they are applied should be relevant.

The final rule is “Do Not Exceed Your Knowledge”. An investigator should not undertake an examination that is beyond their current level of knowledge and skill.

2.1.1.1. Hashing Algorithm

A Message Digest 5 (MD5) is a 128-bit hexadecimal algorithm which can be applied to data, including files and devices. The purpose of an MD5 is to determine the data integrity of a file or device (Bunting, 2008). When a Hard Disk Drive (HDD) is forensically imaged, an MD5 value is generated. This value is generated again when the duplicate digital data is processed. If both values match exactly, then the investigator can verify the integrity of the duplicate image against the original evidence. However, if a different MD5 is generated, then the data of the duplicate image is different to the original HDD.

The likelihood of two MD5 values being exactly the same for two different types of data is highly unlikely. As a result, this is a respected and commonly used method of verifying data.

2.1.2. Forensic Methodology

To ensure forensic soundness, an investigator must follow a process, or methodology, that is reliable, repeatable and documented. There are several forensic methodologies available. Some
methodologies have been revised and updates in order to account for the ever-changing technological world.

2.1.2.1. McKemmish

One of the earliest forensic methodologies is proposed by McKemmish (1999). This forensic methodology encompasses four key elements. These are shown in Figure 2.

![Figure 2. Forensic Methodology Proposed by McKemmish (1999).](image)

Identification refers to the process of knowing what evidence is present, and where and how it is stored is vital to determining which processes are to be employed. The investigator must be able to identify the type of information that is stored in a device and the format in which it is stored so that the appropriate technology can be used to extract it.

Preservation is where the concept of forensic soundness comes into play. Examination and acquisition of electronic data must be carried out in the least intrusive manner to avoid data being altered. In circumstances where alteration of electronic evidence is inevitable, the investigator should be able to explain what data has changed and why. Any alteration to electronic data should be accounted for and justified.

Analysis is regarded as the main element of forensic computing. Once acquired from a device, digital evidence will require processing. Processing allows digital evidence to be extracted in a format that is readable by the investigator so that evidence can be interpreted.

Finally, presentation refers to the evidence being presented in a Court of Law. This is usually in the form of a report which will explain all findings and how or why they may be relevant to an investigation.

2.1.2.2. An Abstract Digital Forensics Model

Reith, Carr and Gunsch (2002) present the Abstract Digital Forensics Model. This methodology is inspired by the Digital Forensic Research Workshop model originally proposed by Palmer (2001), and previous forensic methodologies. There are nine key elements. These are shown in Figure 3.
Figure 3. An Abstract Digital Forensics Model (Reith, Carr and Gunsch, 2002).

Identification refers to recognising an incident and determining the type. Reith et al. recognise that this is not explicit in the field of forensics, but it can have an impact on other steps. Preparation refers to the process of preparing the necessary tools and techniques. A search warrant may be required in some instances. Approach Strategy refers to the process of formulating an approach to collecting data. In this case, the impact on victims and the technology in question should be considered. The aim is to maximise the collection of evidence while minimising impact to the victim. Preservation is one of the most important stages. This refers to the process of isolating, securing and preserving the state of the evidence, both physical and digital. This includes protecting an electronic device from potential external threats, such as electromagnetic interference. Collection involves recording the scene using standardised procedures. An investigator may take detailed notes or take photos of the scene to refer back to later. Examination is the process of performing an in-depth systematic search of evidence. The investigator should identify evidence related to the suspected crime. All findings should be logged in detail. Analysis involves determining the significance of the data found. This involves reconstructing fragments of data and drawing conclusions based on these findings. Presentation refers to the process of summarising the findings and providing an explanation of how and why an investigator has reached a conclusion. This should be written in layperson’s terms. Finally, Returning Evidence involves ensuring that any physical or digital property is returned. Reith et al. recognise that this phase is rarely addressed in many other methodologies.

Maintaining forensic soundness during an investigation is absolutely essential. Most situations allow for a duplicate copy of original data to be achieved. However, as discussed, there are
situations where this is not possible. Forensic soundness should still be maintained with a detailed documentation of how and why data is altered.

2.2. Unmanned Aerial Vehicles

2.2.1. Classification of UAV Systems

There are four ways that a UAV can be categorised. This is dependent on size, make, model, features, and capabilities.

A fixed-wing UAV system is one that uses fixed wings in combination with forward airspeed to generate lift, like an aeroplane (Custers, 2016). A fixed wing system need only exert energy to move forward. As they can fly at higher altitudes, they are able to cover longer distances, map larger areas, and carry heavier payloads. They are, however, financially expensive and restricted in movement compared to other UAV systems. (Australian UAV, n.d.). Fixed-wing systems vary in size and design. They are typically used for purposes such as aerial mapping, construction, security, and surveillance (Olson, 2017). An example of a consumer fixed wing system is the Parrot DISCO Fixed Wing Drone. This is currently available on Amazon for £349.00. This UAV offers First Person View (FPV) goggles, a full HD 1080p camera, and up to 45 minutes of flight time (Amazon, n.d.).

The multirotor system is a subset of the rotorcraft. The term ‘rotorcraft’ is used to define an aircraft which use rotary wings to generate lift. Rotorcraft systems are almost always equipped with multiple small rotors which are necessary for their stability (Custers, 2016). The multirotor system can be further categorised depending on the number of rotors equipped. For example, the tricopter (three rotors), quadcopter (four rotors), hexacopter (six rotors) and octocopter (eight rotors), are among those available. Multirotor systems, specifically quadcopter systems, are popular amongst consumers and business. They offer advantages such as easy control, hovering capabilities and the ability to take off and land vertically. On the other hand, many multirotor systems have a very limited flight time and most multirotor systems do not support a heavy payload (Olson, 2017). There is a direct correlation between price and feature availability. Multirotor systems can be used for photography, agriculture, security, and more. The price is dependent on what an individual or business plans to use the UAV for (Olson, 2017).

At the time of writing, the DJI Mavic Pro is considered one of the best multirotor UAV systems on the market. It is priced at £1099 and is available direct from DJI. This quadcopter offers
incredible 4K and HD footage, control and distance coverage of up to four miles of continuous flight, and close to 30 minutes of flight time (Adams, 2018). The AMZtronics T20CW is the Amazon’s Choice for a budget photography UAV. It is available for £69.99. This quadcopter offers 720p HD capture but has a limited flight time of fewer than 15 minutes (Amazon, n.d.). The Parrot AR Drone 2.0 GPS Edition is also part of the quadcopter subset. It is currently available on Amazon for £407.82 (Amazon, n.d.).

A single rotor system has only a single rotor, plus a tail rotor to control the direction and maintain stability. A single rotor UAV system is strong, stable and can carry heavier payloads. They are however financially expensive to maintain. These single rotor systems are typically used for surveying and scanning purposes (Olson, 2017).

The Hybrid Vertical Take Off and Landing (VTOL) systems are the newest development of UAVs. VTOL systems characterise multirotor and fixed wing systems, using multiple rotors to take off and land, but also fixed wings to fly longer distances (Custers, 2016). VTOL systems take the positives of each separate system and combine them into one, offering flexibility and the ability to perform various manoeuvres that are not possible with a conventional fixed wing system.

### 2.2.2. Basics of a UAV Multicopter

A UAV may be built differently depending on the manufacturer or the model, include different features, or run one of the many OS’s available. However, the internal components included in a multirotor system, especially a quadcopter, are usually quite standard. Figure 4 represents the typical internal components of a UAV quadcopter.
1. The frame is responsible for carrying all of the components of the UAV. Most UAV frames are made from carbon fibre as it is light and strong (BeginnerFlyer, n.d.).

2. The propellers and motors provide the UAV with lift, speed and direction. Propellers can be made from different materials such as carbon fibre and wood (Juniper, 2016).

3. The Flight Control Board (FCB) is considered the brain of a UAV. It is responsible for components including the Electronic Speed Controller (ESC), the accelerometer, and the GPS sensor (Juniper, 2016).

4. A Lithium Polymer (LiPo) battery is usually the choice for providing power to the UAV as they are small, light and can hold a lot of power. The LiPo battery connects to the Power Distribution Board (PDB).

5. The PDB is responsible for distributing the correct amount of power to each component of the UAV such as the ESC and camera (BeginnerFlyer, n.d.).

6. The ESC distributes power to each individual motor according to the signals and instructions sent by the FCB (Juniper, 2016).

7. Commands issued from the controller are received by the receiver component (Juniper, 2016). The controller may be a mobile phone device with the flight application installed, or a genuine controller specific to a UAV.

8. The Global Positioning System (GPS) module provides navigation and positioning data to the FCB (BeginnerFlyer, n.d.).
9. A camera isn’t a necessary component, however, many UAV’s in the market do include at least one camera. The camera is powered by the PDB and, together with a transmitter, transmits a live video signal to the controller (Juniper, 2016).

It is important to note that commercial and hobbyist users can make configurations to their UAV. They may add, remove or replace components to configure the UAV for their use.

### 2.3. Parrot AR Drone 2.0 GPS Edition

The Parrot AR Drone 2.0 GPS Edition is manufactured by the French company Parrot. It is controlled using the AR Free Flight application, which can be installed on a mobile phone or tablet device running either Android or iOS.

The Parrot AR Drone 2.0 was first unveiled at the 2012 CES in Las Vegas and is the next generation of the original Parrot AR Drone unveiled at the same event in 2010 (Hodgkins, 2012). The Parrot AR Drone 2.0 came in three editions including the Power Edition, the Elite Edition and the GPS Edition, each with its own features. At the time of writing this, the Parrot AR Drone 2.0 is no longer available for purchase from the online Parrot store but can still be purchased from other retailers such as Amazon.

#### 2.3.1. Technical Specifications

The Parrot AR Drone 2.0 processor is based upon a 32-bit ARM Cortex A8 running at 1GHz. The memory includes 1GB DDR2 RAM at 200MHz. The Parrot AR Drone 2.0 runs Linux 2.6.32. For Wi-Fi, the Parrot AR supports standards b/g/n. For a flight, the Parrot AR includes 3 axis accelerometer (+/- 50mg precision), a 3 axis gyroscope (2000 degree precision) and 3 axis magnetometer (6 degree precision). For height, a pressure sensor (+/- 10 Pa precision) and ultrasound sensors for ground altitude measurement are used. A vertical QVGA at 60fps is used for ground speed measurement. A USB 2.0 connection is included for extension (Parrot AR Drone 2.0, 2012).

Expanded Polypropylene (EPP) is used for the hull. Motors are mounted on to Carbon Fibre tubes weighing 380g for the outdoor hull and 420g for the indoor hull. A LiPo rechargeable battery at 1000 mA/H provides power to the Parrot AR Drone 2.0 (Parrot AR Drone 2.0, 2012). The front camera offers HD 720p at 30fps with a wide-angle lens (92 degree diagonal). Video and photo data can be stored on a connected USB or the control (Parrot AR Drone 2.0, 2012). Parrot AR Drone 2.0 parts can be purchased online from the Parrot store individually.
2.3.2. Parrot Mobile Application

The AR Free Flight application is the current application used to pilot the Parrot AR Drone 2.0. The direction of the Parrot AR Drone 2.0 is controlled by tilting the mobile phone or tablet control device. Users can also perform tricks, take photos or record HD video (Google Play, n.d.).

AR Free Flight is currently available for both iOS and Android and can be downloaded from either the App Store or Google Play. The current version available from Google Play is AR.FreeFlight 2.4.15. This application requires Android 2.2 and up. The application was last updated August 2016 and has been installed over 1,000,000 times (Google Play, n.d.). Upon launching the AR.FreeFlight 2.4.15 application, the user is presented with the interface shown in Figure 5. The user can select the required option.

![AR.FreeFlight Interface](image)

Figure 5. AR.FreeFlight.2.4.15 Home User Interface (Google Play, n.d.).

Upon tapping the “Piloting” option, the user is presented with the interface shown in Figure 6. The camera of the Parrot AR Drone 2.0 will emit a live video feed to the flight controller. This is where the user can control the direction of the Parrot AR Drone 2.0 and take advantage of features such as taking photo and video.
Figure 6. AR.FreeFlight 2.4.15. Piloting User Interface (Google Play, n.d.).

AR Race 2 is an application multiplayer racing game which allows users to race against each other to finish in the fastest time. AR Race 2 is currently only available for download from the App Store for iOS devices (App Store, n.d).

2.3.3. Flight Recorder

A USB-like Flight Recorder is included with the Parrot AR Drone 2.0 GPS Edition as standard. The Flight Recorder includes 4GB of memory and offers new features, including geolocation, improved stability, destination point selection, flight recording and video (Parrot AR Drone 2.0, 2012). Other features of the Flight Recorder include:

- Return Home: The Parrot AR Drone 2.0 returns to its take-off point in a straight line by itself video (Parrot AR Drone 2.0, 2012).
- Control by Map: A new mode which allows the user to choose the destination of the Parrot AR Drone 2.0. A new control interface shows a geographical map on which a user can tap to determine the destination of the UAV video (Parrot AR Drone 2.0, 2012).
- AR Drone Academy: Flight data (altitude, speed, etc.) is enriched by GPS location data. By registering with the AR Drone Academy, a user can access all of the flights made and review the trajectory in 3D position on the AR Drone Academy map video (Parrot AR Drone 2.0, 2012).
Essentially, the Flight Recorder is a black box for the Parrot AR Drone 2.0.

### 2.4. UAV Forensics

As UAV-related incidents continue to increase, it is becoming increasingly important to understand how a UAV operates, what kind of data is stored, and how this data is valuable in an investigation. It is important to understand the different OS’s that a UAV can operate as each one can present individual challenges. Several studies have been carried out which examine and analyse other UAV models and manufacturers. These studies highlight some of the challenges facing digital forensic investigators when it comes to UAV forensic analysis.

#### 2.4.1. UAV Methodology

Jain et al. (2017) proposed a 12-phase forensic framework to help provide a systematic approach to the investigation of UAV’s. This 12-phase framework can be seen in Appendix A. Five commercial UAVs, including the Parrot AR Drone 2.0, were examined extensively to identify and understand the relevance of the various components.

To validate their framework, an experiment was performed. Modifications, such as adding and removing components, were made to each UAV. This would help determine whether the proposed framework included all components present in any basic commercial UAV and would, therefore, be applicable to perform a complete UAV analysis.

In their conclusion, Jain et al. argue that the lack of law enforcement training procedures in the field of UAV’s is one of the biggest roadblocks in mitigating attacks. Although Jain et al. did not perform a forensic analysis of any of the five UAV’s they examined, they provide a useful framework to aid investigators in the examination and analysis stages.

#### 2.4.2. DJI Forensics

Clark et al. (2017) presented the first comprehensive analysis of the DJI Phantom 3 Standard. They also developed a forensically sound open source DRone Open source Parser (DROP) tool.

The UAV was flown in two geographical locations. Data acquisition was then broken down into three parts: drone, controller, and phone/tablet.

Clark et al. identified two files of interest. The .txt files which were created by the DJI GO application and the .dat files created by the UAV. After decrypting and decoding these files, flight information related to GPS locations, Wi-Fi connections, motors, remote control, flight
status, and other information was extracted. Once the data was analysed, and the proprietary file structures were understood, Clark et al. developed the DROP tool which they used to parse the evidentiary files. Clark et al. found that turning on the UAV affected the integrity of the data stored on its internal storage. Each time the UAV was turned on a new .dat file was created. They also noticed that if the SD card was at or near full capacity, turning the UAV on immediately wiped the oldest data in an unrecoverable manner. Clark et al. concluded that although their study provides a good starting point for UAV forensic analysis, more work is needed to cover the wide spectrum of UAV’s available for consumers today.

Maarse and Sangers (2016) presented a forensic analysis of the DJI Phantom 2 Vision Plus. The aim of their study was to answer the question “Can the flight path of a UAV be reconstructed using positional data gathered from a UAS?”. They also briefly investigated counter forensic methods and whether these could be detected.

An acquisition of components, including the DJI Vision application, the UAV and range extender, and the UAV’s onboard camera and Micro SD card, was performed in both pre-flight and post-flight conditions. Maarse and Sangers executed a small flight plan on University premises. They identified artefacts related to the UAV in memory dumps of the DJI Vision application. This included coordinates of the most recently added waypoint, coordinates of the UAV’s home point, the UAV’s altitude, and messages of the waypoints being uploaded.

They also identified that image and video files stored on the Micro SD card contain Exchangeable Image File Format (EXIF) data, including latitude and longitude coordinates and time stamp information. In order to verify the integrity of the evidence gathered Maarse and Sangers investigated two possible counter forensic methods. They first found that it was possible to alter media time stamps by altering the system time of the Android flight controller before powering on the UAV. Next, they investigated whether GPS signals could be blocked. They first attempted this by disconnecting the GPS receiver from the main board of the UAV, however, they found that this activates a mechanism which prevents the UAV from taking off. Through further investigation, they found that it is possible to block the GPS signal by simply attaching tin foil directly over the GPS receiver. Consequently, the camera no longer stored GPS attributes in EXIF data, the controller did not record the home point on take-off and blocking the GPS receiver allowed users to fly the UAV in restricted airspace.

Maarse and Sangers concluded that in situations where flight log information is not available, the only information retrieved is EXIF data recorded in media files. They propose that future
work should involve using similar methods for investigating newer models, such as the DJI Phantom 4. They also suggest that an investigation of the new DJI application is carried out using a different mobile platform, such as iOS.

### 2.4.3. Parrot Forensics

Horsman (2016) presented a preliminary forensic analysis of the Parrot Bebop. This is the only UAV which is similar to the Parrot AR Drone 2.0.

Horsman attempted to address the four main challenges of UAV forensic analysis faced by an investigator. The investigation was divided into two parts; an investigation of the UAV, followed by an investigation of the flight controller. Flight data was retrieved from the UAV in the form of ‘.pud’ files. A new .pud file was created each time a session between the UAV and controller was established. A series of metadata was found at the beginning of each .pud file. This metadata contained flight data including the date and time of flight, UAV serial number, flight controller model and flight control application. Images and video taken by the UAV’s onboard camera were identified. Photos maintained EXIF data including the longitude and latitude coordinates of where the images were taken. Establishing ownership was only possible where the UAV and controller have been seized by identifying the serial number.

Analysis of the Android controller found that flight data is recorded and stored in .pud files. The metadata of each .pud file contained the date and time of each flight and the serial number of the UAV flown. Image and video files could be downloaded from the UAV to the flight controller. These files were still recoverable from the internal storage, though subject to being overwritten.

Image files contained EXIF data including the date, time and longitude and latitude coordinates for each image. A ‘UUID’ number was also stored in a .pud file. This documented the flight when an image was taken when paired together. A Extensible Markup Language file (.xml) was stored on the Android flight controller. This file contained information related to the last connected UAV, along with its serial number and longitude and latitude coordinates.

Analysis of the iOS controller (iPhone 6) yielded similar results with flight data being recorded, and media containing EXIF data.

Horsman also found that manipulation of the flight controller’s system time and date was reflected on the UAV. He also found that GPS data could be manipulated. The GPS location was not stored if the location services of the flight controller were turned off. Furthermore, a spoofed GPS location could mask the real location of the UAV. The spoofed location was
embedded in the UAV’s files. Tin foil could also be used to block the GPS signal, which would prevent GPS data from being recorded.

Horsman then performed a factory reset of the UAV. Examination revealed that data was not recoverable, highlighting a further threat and challenge for investigators. Horsman concluded that further analysis across a large test set of UAVs’ is required before best practice guidance can be definitively established. Future work requires the continued examination and presentation of results of different manufacturer devices.

### 2.5. UAV Security

Although a forensic investigation has not yet been done, several studies have highlighted the security vulnerabilities of the Parrot AR Drone 2.0. It is important to understand the vulnerabilities of the Parrot AR Drone 2.0 and discuss how this could have an effect on an investigation.

#### 2.5.1. Parrot AR Drone 2.0 Vulnerabilities

Pleban et al. (2014) aim to describe the security vulnerabilities of the Parrot AR Drone 2.0 and discuss how it can be secured from unauthorised access.

A port scan identified two open and well-known ports. File Transfer Protocol (Port 21) is not password protected and provides open access to the /data/video subdirectory of the UAV. A USB device connected to the UAV is mounted at /data/video/usb. Direct access to the USB is possible. An attacker may use this vulnerability to inject malicious files onto the UAV or USB or gain access to modify, add or delete files. Telnet (Port 23) leads to a root shell. The root account is not password protected, giving an attacker open access to the entire OS. An attacker can take advantage of this vulnerability by performing malicious activities. Such activities include changing important configuration files, wiping the filesystem, or resetting the drone mid-air resulting in the motors being turned off. Since the Parrot AR Drone 2.0 runs on Linux, the same attacks used to attack a computer can be applied to the UAV.

An attacker can gain full backdoor access to the UAV using these methods of access by changing the reset script. For example, the shell script /bin/reset_config.sh shows that after pressing the reset button on the drone, only the config file /data/config.ini is reset. Other files remain the same. This allows an attacker access to the drone, even after a user has reset it. The only security feature offered by Parrot AR Drone 2.0 is a MAC address filter. A user can authorise connections through the MAC address and any non-authorised access is blocked. This
isn’t sufficient as a MAC address can be easily spoofed and passed off as an authorised connection.

Samland et al. (2012) presented a similar security threat analysis with a similar aim of sensitising the possibility of misuse and the motivation for an implementation of improved security mechanisms. They identify that the main security vulnerability is the communication over an unencrypted connection and the use of User Datagram Protocol (UDP). They recreate three scenarios where the Parrot AR Drone 2.0 could be misused or at risk to an unauthorised individual.

In their first scenario, they attempt to high-jack the drone. Connecting to the UAV is simple as the network is not secured. Access to the OS is possible through Telnet. An attacker can then determine the MAC and IP address of the legit flight controller. The attacker can alter the MAC address of the non-authorised device and gain control of the UAV. The SSID of the UAV’s network can also be altered, resulting in the immediate disconnection of the flight controller. The attacker can then connect to the new SSID of the drone.

In their second scenario, they attempt to intercept the video signals of the AR Drone. File Transfer Protocol (FTP) is unsecured. An attacker can take advantage of this by transferring and executing malicious code to the UAV. Samland et al. execute a malicious code, which is intended to intercept the video stream of the UAV and send it to the attacker while avoiding suspicion.

In their final scenario, they attempt to track a person and determine their position using the drone and GPS receiver. This involves first fitting a smartphone with a GPS receiver to the drone. The drone is controlled manually using a PC. In order to integrate the smartphone into the communication cycle, a proxy is developed for the drone’s communication. The proxy is located on the smartphone and communication flows from the PC to the smartphone, to the drone and vice versa. This requires that the smartphone is connected to the same network as the drone and PC, i.e. the network provided by the drone. Communication between the UAV and PC takes place via the proxy phone. The proxy provides threads that monitor open ports to communicate and pass the information to the remote site. The PC connects to the proxy, which connects to the drone. Thus, the actual user device is the phone. The GPS server of the proxy follows the principle that the user device reports to the open ports of the drone in which information is sent and received. By extending a control program to open an additional connection to the provided port of the GPS server, the proxy will send the attacker the determined GPS coordinates and current time using this connection.
Peacock (2014) further verified the lack of security over Port 21 (FTP) and Port 23 (Telnet) but also identified certain TCP ports can be used to identify a Parrot device. For example, Port 5551, 5553, 5555, 5557 and 5559 all have Parrot specific purposes. Multiple devices can connect to the Parrot AR Drone 2.0 at any one time. However, if multiple devices attempt to interact with any data streaming from or to the drone, such as video stream or control commands, the devices would seize up. With the video stream, when one device attempts to connect while another is currently connected, the original device is frozen, and the newly connected device receives the stream. Peacock theorised that a Deauthentication attack could be used against the Parrot to disconnect the device and controller. Having successfully disconnected the controller from the drone using aircrack-ng to send Deauthentication packets, the drone would enter generic states. For example, losing the control data link resulted in “hover mode” whereby current altitude is maintained until a connection is re-established. If re-established, the Parrot enters “emergency landing mode”, whereby the Parrot lands before control is completely regained.

### 2.6. UAV Legislation

The Civil Aviation Authority (CAA) is responsible for regulating airspace in the United Kingdom (UK) (Gov UK, n.d.). The CAA base their regulations on the size and weight of a UAV. According to the CAA, a Small Unmanned Aircraft is one which weight 20kg or less. This includes all types of traditional remotely control model aeroplanes, helicopters, or gliders, as well as multirotor ‘drones’ and remotely controlled ‘toy’ aircraft (Civil Aviation Authority, 2015). The Air Navigation Order (ANO) 2016 regulates aviation safety standards and aircraft navigation which fall outside of European Union (EU) regulation. It covers areas including aircraft, aircrew, passengers, cargo, air traffic services and aerodromes.

According to The Air Navigation Order (Part 10., Ch. 1., ss. 241.). “A person shall not recklessly or negligently cause or permit an aircraft to endanger any person or property”. This applies to all aircraft, including UAV, regardless of weight, size or category (The Air Navigation Order 2016., Part 2., Ch. 1., ss. 23.).
2.6.1. Small Unmanned Aircraft

The Air Navigation Order 2016 (Part 5, Ch. 4., ss. 94) sets out regulations of a Small Unmanned Aircraft as follows:

(1) A person must not cause or permit any article or animal (whether or not attached to a parachute) to be dropped from a small unmanned aircraft so as to endanger persons or property.

(2) The person in charge of a small unmanned aircraft may only fly the aircraft if reasonably satisfied that the flight can safely be made.

(3) The person in charge of a small unmanned aircraft must maintain direct, unaided visual contact with the aircraft sufficient to monitor its flight path in relation to other aircraft, persons, vehicles, vessels and structures for the purpose of avoiding collisions.

(4) The person in charge of a small unmanned aircraft which has a mass of more than 7kg excluding its fuel but including any articles or equipment installed in or attached to the aircraft at the commencement of its flight, must not fly the aircraft—

   (a) in Class A, C, D or E airspace unless the permission of the appropriate air traffic control unit has been obtained;

   (b) within an aerodrome traffic zone during the notified hours of watch of the air traffic control unit (if any) at that aerodrome unless the permission of any such air traffic control unit has been obtained; or

   (c) at a height of more than 400 feet above the surface unless it is flying in airspace described in sub-paragraph (a) or (b) and in accordance with the requirements for that airspace.

(5) The person in charge of a small unmanned aircraft must not fly the aircraft for the purposes of aerial work except in accordance with a permission granted by the CAA.
2.6.2. **Small Unmanned Surveillance Aircraft**

The Air Navigation Order (Part 5., Ch. 4., ss. 95) sets out regulations of a Small Unmanned Surveillance Aircraft as follows:

(1) The person in charge of a small unmanned surveillance aircraft must not fly the aircraft in any of the circumstances described in paragraph (2) except in accordance with permission issued by the CAA.

(2) The circumstances referred to in paragraph (1) are:

(a) over or within 150 meters of any congested area;

(b) over or within 150 meters of an organised open-air assembly of more than 1,000 persons;

(c) within 50 meters of any vessel, vehicle or structure which is not under the control of the person in charge of the aircraft;

(d) subject to paragraphs (3) and (4), within 50 meters of any person.

(3) Subject to paragraph (4), during take-off or landing, a small unmanned surveillance aircraft must not be flown within 30 meters of any person.

(4) Paragraphs (2)(d) and (3) do not apply to the person in charge of the small unmanned surveillance aircraft or a person under the control of the person in charge of the aircraft.

(5) In this article ‘a small unmanned surveillance aircraft’ means a small unmanned aircraft which is equipped to undertake any form of surveillance or data acquisition.

2.6.3. **Future Legislation**

In 2017, the Department for Transport announced several new regulations regarding UAVs. Under these new regulations, UAV’s weighing 250 grams or more are to be registered. There is currently no registration process in place in the UK. The Federal Aviation Association (FAA) are responsible for the regulation and oversight of civil aviation within the United States of
Currently, the FAA operates a registration process which requires a UAV to be registered if it weighs more than 0.55lbs (250 grams) and less than 55lbs (25kg) (Federal Aviation Administration, n.d.).

A mandatory Drone Safety Awareness test will also be introduced. Owners will need to prove that they understand UK safety, security, and privacy regulations before being allowed to operate a UAV (Department for Transport, 2017).

Work to create an authoritative airspace will also be brought forward whilst simultaneously bringing forward ‘geo-fencing’. This will act as an invisible shield around unauthorised and restricted areas and prevent a UAV from entering such zones (Department for Transport, 2017).

Finally, further measures such as increasing penalties, creating new offences and reviewing powers available to law enforcement to enforce relevant laws and legislation will be explored. These changes are expected to feature in the Spring 2018 government bill (Roberts, 2017).

This legislation attempts to regulate UAVs by placing certain restrictions on them. By restricting the maximum altitude a UAV can reach, a UAV should not collide or cause an issue with other aircraft such as aeroplanes. By restricting the area that an individual can fly a UAV such as within a certain distance of a congested area, or within a certain distance of a structure, the opportunity to carry out criminal offences such as those discussed in Section 1.1, such as spying, should be mitigated. However, as discussed this is not the case. As a result, UAV legislation is constantly being reviewed and updated.

2.7. Mobile Controller

An Android mobile phone device will act as the flight controller in this project. A discussion of mobile phone devices is presented in this Section. A discussion of the Android OS is made in Section 2.9.

Mobile phone devices have advanced significantly in the last 20 years. Earlier mobile phone devices were restricted to perform basic features such as making and receiving a call. Newer devices, often referred to as ‘Smartphones’, provide additional functionality and features such as Internet browsing, photos and video, GPS, and much more. Smartphone devices have essentially become a compact version of the once popular personal computer (PC) with high performance, enhanced functionality, and their own mobile OS. As a result, smartphone devices have become an integral part of our everyday lives. According to Statista (2016), there were
1.57 billion smartphone users in the world in 2014. This figure is projected to rise to 2.87 billion by 2020.

Modern mobile OS’s combine the features of a mobile phone device with that of a PC, whilst also providing an environment for other programs to run (Khurana, 2014). Symbian, a mobile OS developed by Nokia, was the first popular mobile OS in the world accounting for 48.8% of the global mobile OS market in 2009. In the same year, Blackberry’s Research in Motion (RIM) followed behind accounting for 20.6% of the market. Despite the early and clearly successful start, both Symbian and RIM have been driven out of the current market. Google’s Android has been steadily climbing since 2009 and is currently the most popular mobile OS in the world accounting for 87.7% of the market. Apple’s iOS follows accounting for 12.1% of the market (Statista, 2017). This is shown in Figure 7.

![Figure 7. Global Mobile OS Market Share 2009-2017 (Statista, 2017).](image)

## 2.8. Mobile Forensics

As discussed in Section 2.1, the term digital forensic investigation can have many branches which are related to this field. Mobile forensics is a branch of digital forensics which relates to the acquisition and analysis of digital evidence from a mobile phone device (Tamma and Tindall, 2015). As individuals grow increasingly reliant on the use of their mobile phone devices for everyday personal use, the trail of digital information continues to grow. There is a
wealth of forensically relevant data residing on a mobile phone device which can be extracted by a digital investigator, including call history, Internet history, text and multimedia messages, image and video, application data, and more.

2.8.1. Mobile Forensic Challenges

Accessing and extracting this wealth of forensically valuable data is not without its challenges however and a digital investigator must be cautious when attempting the data acquisition of a mobile phone device.

Data is dynamic and highly volatile. Even if a device appears to be off, background processes may still be running. For example, in most mobile phone devices, the alarm clock will still work even when the phone is off. A sudden transition from one state to another may result in data loss or modification (Mahalik, 2016).

Keeping with the changes in one mobile OS is challenging enough, however, keeping up with multiple OS’s across multiple mobile phone manufacturers is nearly impossible. Updates in OS are frequent and major updates are usually released every quarter. As such, there is no single and generic tool or method which is sufficient and applicable to the forensic acquisition of all mobile phone devices (Tahiri, 2016).

The hardware architecture of a mobile phone device varies across manufacturers also. A mobile phone device will differ in size, model, hardware, features and OS. A digital investigator will need hundreds of adapters and power cords depending on the device and hardware (Tahiri, 2016).

Security features are built in to protect user data and privacy. Smartphone devices offer protection for individual files or directories. Some new mobile OS’s even offer full disk encryption (Tahiri, 2016).

Wiped data cannot be recovered easily. Most modern smartphone devices offer various methods of wiping data. For example, Android devices can be wiped via Google Sync, iPhone devices can be wiped via iCloud. Data can be wiped via a desktop manager or upon entering a password wrong a certain number of times (Tahiri, 2016).

These are just a few of the many challenges facing digital investigators in the field of mobile forensics.
2.8.2. Mobile Device Tool Classification System

Originally developed by Brothers et al. (2014), the Mobile Device Tool Classification System provides a framework for digital investigators to classify and compare the extraction methods of different tools. The Mobile Device Tool Classification System is shown in Figure 8. The tools and methods become more forensically sound as the pyramid progresses upwards, however, they also become more complex and require longer analysis time.

![Mobile Device Tool Classification System Diagram](image)

Figure 8. Mobile Device Tool Classification System (Brothers et al., 2014).

At Level 1 of the pyramid is the Manual Extraction. A Manual Extraction involves scrolling through the data on the device and viewing the data directly using the device’s keypad or touchscreen. Actions taken by the digital investigator may be video recorded. Data of forensic value may be photographed. Deleted data is impossible to recover at this level (Brothers et al., 2014).

At Level 2 of the pyramid is the Logical Extraction. A Logical Extraction involves connecting a device to a forensic workstation by either Universal Serial Bus (USB), RJ-45, Wi-Fi or Bluetooth. Different connection types and protocols may result in data on the device being modified. Once connected, a series of commands are sent from the forensic workstation to the device. The device responds by sending the requested data to the forensic workstation (Brothers et al., 2014).

At Level 3 of the pyramid is Hex Dumping/ Joint Test Action Group (JTAG). These methods of extraction provide a digital investigator with direct access to raw data stored in flash memory. Hex Dumping involves uploading modified software into a protected area of memory. This is accomplished by connecting a device’s data port to a flasher box and the flasher box to a forensic workstation. The device is placed in diagnostic mode by sending a series of commands from the flasher box. The flasher box can then capture all, or portions, of flash memory and send it to the forensic workstation (Brothers et al., 2014).
JTAG involves connecting to the Test Access Ports (TAPs) on a device. The device then transfers raw data from memory chips to the forensic workstation via the device’s microprocessor. JTAG is more invasive than Hex Dumping as it usually involves dismantling the device to gain access to connections (Brothers et al., 2014). At Level 4 of the pyramid is Chip-Off. A Chip-Off extraction involves acquiring data directly from a device’s flash memory. This involves the physical removal of flash memory through a process of de-soldering. Data extracted is presented in a binary format. A digital investigator must then parse, decode and interpret this data into a readable format (Brothers et al., 2014). Finally, at Level 5 of the pyramid is Micro-Read. A Micro-Read extraction involves using an electron microscope to record and analyse the physical observation of the gates on a NAND or NOR chip (Brothers et al., 2014).

2.9. Android

Android is an open source mobile device platform developed on top of the Linux 2.6 kernel. Android was originally developed in 2003 by Android Inc. Google purchased Android Inc. in 2005. The T-Mobile G1, also known as the HTC Dream, was released in 2008. This was the first phone to run Android version 1.0 (Hoog, 2011).

Google’s Android is currently the most popular mobile OS in the world accounting for 87.7% of the market (see Figure 7). In 2009, 6.8 million Android smartphones were sold. By 2015, this figure had risen to more than 1.16 billion (Statista, 2017). It is predicted that by 2021, 85.3% of mobile devices will run Android (Statista, 2017). This is shown in Figure 9.
2.9.1. **Android System Architecture**

The Android OS is responsible for managing system resources and providing a way for applications to communicate with hardware and software components. The Android OS consists of a stack of layers running on top of each other. Each layer provides services to the layer above it. The architecture of the Android OS is shown in Figure 10.

![Market share of worldwide smart OS from 2014 to 2022](image-url)
Figure 10. Android System Architecture (Android, 2018).

The Linux Kernel is what the entire Android OS is built on. It provides a way of communication between the hardware and software of a device. Core functionalities including process management, memory management, security and networking, are managed by Linux kernel (Tamma and Tindall, 2015).

Android’s native libraries are written in C or C++ and help the device handle different types of data. For example, Media Framework supports the recording and playback of audio, video and picture format (Tamma and Tindall, 2015).

On the same layer as the Libraries is the Android Runtime which runs the Dalvik Virtual Machine (DVM). Android applications are programmed in Java. When compiled, a Java program outputs a bytecode which can be executed by a Java Virtual Machine (JVM). However,
for Android, the Java bytecode is further converted to Dalvik bytecode by a dex compiler. The Dalvik bytecode can then be executed by the DVM. Each Android application runs its own instance of the DVM. Dalvik bytecode is used as a suitable alternative for low-memory and low-processing environments (Tamma and Tindall, 2015).

The Application Framework layer is responsible for resource management, location management, handling calls, device activity, and more (Tamma and Tindall, 2015).

Finally, the Applications layer consists of programs which users can directly interact with.

System Applications are those that are pre-installed on the device. These cannot be changed or uninstalled by the user. These applications are usually mounted in the /system directory.

User Applications are those downloaded and installed by the end user from platforms such as Google Play. These applications are usually mounted in the /data directory (Tamma and Tindall, 2015).

2.9.2. Android Boot Process

The Android boot process loads the firmware, OS and user data into memory to support the operation of a device. The firmware and Read Only Memory (ROM) varies by manufacturer and device. There are seven steps to the Android boot process:

1. Boot ROM Execution: When first powered on, a boot ROM code is paired with the Central Processing Unit (CPU) and executed to first initialise hardware, and then locate the boot media. This is similar to the Basic Input-Output System (BIOS) process used to boot a PC. Once the boot media is located, the initial boot loader is copied to internal Random Access Memory (RAM) (Hoog, 2011).

2. Boot Loader: The boot loader has two initial stages: the initial program load (IPL) and the second program loader (SPL) (Hoog, 2011).

   The IPL is responsible for the detection and set up of external RAM. Once external RAM is prepared, the IPL copies the SPL into RAM and execution is transferred.

   The SPL is responsible for loading the Android OS and providing access to alternative boot modes such as fast boot and recovery. Typically, the SPL will initialise hardware components such as the console, display, keyboard, file systems, and other features required to operate the device. The SPL then transfers execution to the Linux kernel (Hoog, 2011).

3. Linux Kernel: As acknowledged, the Linux kernel is the heart of the Android OS. It is responsible for process management, memory management, security, and networking.
The root file system is read from the flash memory. This will provide access to system and user data (Hoog, 2011).

4. **The Init Process:** This is the very first process. The init.rc is typically located on the root file system. The init scripts provide details on how to start key system and user processes. At this stage, the user will see the Android logo on the display of the device (Hoog, 2011).

5. **Zygote and Dalvik:** The Zygote sequence sets up the Java runtime environment and initialises the Dalvik Virtual Machine. Without the Zygote service, applications would not operate (Hoog, 2011).

6. **System Server:** The core features such as telephony, network and other fundamental components that the device and its applications rely on are executed from here. Once the boot process is complete, the system server sends a broadcast action called ACTION_BOOT_COMPLETED. This alerts dependent processes that the boot process is complete (Hoog, 2011).

7. **Complete:** The Android device is fully operational and is ready for interaction with the end user (Hoog, 2011).

### 2.9.3. Android File System

Android supports file systems including Fourth Extended Filesystem (EXT4), File Allocation Table (FAT32) and Yet Another Flash File System (YAFFS2). The YAFFS2 file system was originally used for Android devices. Android switched to an EXT4 file system in 2010 with the release of Gingerbread (Version 2.3) as YAFFS2 “would have been a bottleneck on dual-core systems” (Mahalik, 2016).

Android uses partitions to structure and store data. Certain folders in an Android system are only available through root access. The term “root” refers to the default administrative account. The root user can start and stop processes, edit or delete any file, change the privileges of other users, and more.

- **/boot:** Contains the information and files required to boot a device including the boot loader and Linux kernel. This partition cannot be modified (Cinar, 2015).
- **/system:** Contains Android system files and applications preloaded by the manufacturer.
- **/recovery:** Contains a backup image of the device and allows the device to boot into recovery mode for repair and restore of the system (Cinar, 2015).
• /data: Contains the data of each application on the device. This partition has significant forensic value but is protected by system permissions (Cinar, 2015).

• /cache: Contains frequently accessed and temporary files. On most Android devices, this partition is stored in RAM. On device power down, the content of this partition is lost (Cinar, 2015).

• /misc: Contains miscellaneous settings regarding the state of the device and information about the hardware settings, USB settings, and more (Mahalik, 2016).

• /sdcard: Contains all of the information present on the SD card including pictures, videos, files, documents etc (Mahalik, 2016).

2.9.4. Android Forensic Data Acquisition

There are many considerations which a digital investigator must consider before acquiring data from an Android device. Most acquisition methods require USB Debugging to be enabled, which can present a challenge if the device is locked. Root access to the device is required in some cases to acquire certain data from a device. A digital investigator must consider a suitable method of data acquisition and how the integrity of mobile data could be affected. There are three types of data acquisition methods available for an Android device, manual, logical, and physical. Some of these methods of acquisition have overlap with those discussed in Section 2.8.2.

Manual acquisition involves using the keypad or touchscreen of a device to navigate through the interface and view the data directly from the device. A digital investigator may video record the actions taken. Evidence of forensic value is photographed and documented. This method of data acquisition is available for most devices except in cases where a device is locked/encrypted or damaged to a point that a digital investigator cannot use the touchscreen or navigate through the interface. Where and if possible, a logical or physical acquisition are much more suitable alternatives as data acquisition methods.

Logical acquisition involves acquiring data on the device by accessing the file system. Various factors, such as whether a device can be rooted, will determine how much forensically valuable data can be acquired from a device. For example, having root access to a device will allow a digital investigator full access to partitions such as /data which is otherwise not accessible. Furthermore, an investigator should be aware that rooting a device modifies data. Logical acquisition can be performed in three ways:
1. The dd command to SD card: The dd command can be used to create a raw bit-by-bit image of a device. This method involves first rooting the Android device with the device connected to a forensic workstation. USB Debugging should also be enabled on the device and the Android SDK should be installed on the forensic workstation. The “adb shell” command provides access to the device. An SD card can then be mounted. This is where the device image is saved. A digital investigator can identify the partition and block they wish to image. The command “dd if=/dev/block/<block number> of=/sdcard/<image name>” will create a bit-by-bit copy of the block or partition. The saved image can then be analysed using forensic software (Mahalik, 2016).

2. ADB Pull: This is similar to the method above. Again, this method involves first rooting the Android device with the device connected to a forensic workstation. USB Debugging should also be enabled on the device and the Android SDK should be installed on the forensic workstation. The “adb shell” command provides access to the device. The “mount” command will provide the device partitions and blocks. The command “adb pull /<partition> <area of image to save to>” will create a bit-by-bit copy of the device partition or block and store this to the file path specified. For example, “adb pull /data C:\AndroidForensics”. The application data stored in /data is copied to a folder called ‘AndroidForensics’ on the C: of the digital investigator’s forensic workstation (Mahalik, 2016).

3. Content Provider: This method can be used on either a rooted or non-rooted device, however, USB Debugging must be enabled on the device. During installation of an application, a user determines whether or not the application can have access to certain data such as contacts, calendar, etc. Without this permission, one application cannot be accessed by another application. Content providers present data to external applications in the form of one or more tables. A digital investigator can take advantage of this by creating an application that can acquire information from the available content providers.

   AFLogical is an application developed by viaForensics which is used to access and extract data from the device. Once installed, the application can be run directly from the device. The AFLogical application will begin to extract data from the chosen content providers. Data is saved as a .CSV file on the SD card of the device (Mahalik, 2016).
Physical acquisition refers to the process of acquiring data in a physically using hardware. This involves two methods:
The Joint Test Action Group (JTAG) method involves connecting to the Test Action Ports (TAPs) of a device and instructing the processor of a device to transfer the stored data. JTAG is invasive, complex, labour-intensive and time-consuming. Also, JTAG doesn’t necessarily bypass encryption (Mahalik, 2016).
Chip-Off involves the physical removal of NAND flash chips from the device through a process of de-soldering. The chip is then inserted into specialised hardware to be read and forensically analysed. Again, Chip-Off is invasive, complex, labour-intensive and time-consuming (Mahalik, 2016).

2.10. Linux

As discussed in Section 2.3.1, the Parrot AR Drone 2.0 runs on Linux 2.6.32. A brief discussion of Linux is presented in this Section. This will provide a better understanding of this OS which could help in the examination of this project.
The Linux OS, modelled after the UNIX OS, was first introduced in 1991 (James, 2011). A Linux Distribution, or ‘Linux Distro’, is a set of components required to make a working Linux system. A Linux Distribution will usually include the Linux Kernel, GNU tools, tools which provide different services, a desktop environment, and a number of software packages (Castro, 2016). Linux and Linux Distributions are free and open-source so a user can choose one depending on their need or preference. For example, Kali Linux is designed specifically for security and penetration testing purposes and Ubuntu is a simple, easy to use and modern OS designed for a large number of users (Newell, 2018). At the centre of every Linux system is the Linux Kernel. The Linux Kernel is responsible for running the entire OS including managing memory and devices, executing commands and handling errors (James, 2011).
Linux can be applied to a number of systems and system uses from mobile phone and smart devices to desktops and servers. As it is an open source platform, Linux is constantly being modified and updated meaning that it is a reliable, robust and stable OS (James, 2011).

2.10.1. Linux System

The Linux OS is made up of three layers as shown in Figure 11.
Hardware includes the main memory as well as one or more CPU’s to perform computation and to read from and write to memory. Devices such as disks and network interfaces are included here (Ward, 2015).

The Kernel is the core of the OS. The Kernel controls the CPU hardware, allocates memory, schedules processes and is responsible for running applications and protecting them from each other. Everything that the Kernel does revolves around main memory. Each system process gets its own share of memory, and the Kernel must ensure that each process keeps its share. These system areas include:

- Processes: Determining which processes are allowed to use the CPU.
- Memory: Keeping track of memory, what is being used and what is free.
- Device Drivers: Act as an interface between hardware and processes
- System Calls and Support: Processes use system calls to communicate with the kernel (Ward, 2015).

Finally, the user space, or user processes, are made up of the running applications that are managed by the Kernel (Ward, 2015).
2.10.2. Linux File System

Linux supports a number of different file systems including EXT, FAT, FAT32, NTFS, VFAT, HPPS and more. This provides a seamless operation of Linux systems in hybrid environments (James, 2012).

Linux uses a hierarchical structure for storing files and directories. At the very top of this hierarchical structure is the *root* directory, which is usually denoted by a forward slash ‘/’. The root directory has a number of directories, which have a number of subdirectories. A subdirectory is a child from the parent directory which it is branched. There are a number of these directories which contain crucial and valuable information relating to the system and the user (EC-Council, 2016).

2.11. Conclusion

This chapter provides in-depth research into the main components included in this investigation. There is limited work related specifically to UAV forensics. However, the previous work discussed in Section 2.4 has provided a sound foundation for the design and examination component of this project. The findings by Horsman (2016) as discussed in Section 2.4.3 are of particular interest as the Parrot Bebop is the closest related UAV to the Parrot AR Drone 2.0. Previous work which has involved investigating counter forensic methods have highlighted the need for this to be investigated in this project also. The only criticism related to this area is the limited investigative research. The work presented here focus on two of the biggest manufacturers in the UAV market, so few comparisons can be made. One of the key concepts highlighted in Section 1.1 is the question of “How can a criminal pilot be identified if only the UAV is found at the crime scene? Or if only fragments from a wreckage are found? Or when only the flight controller or UAV is found?” (Marks, 2017). These are the questions that will be kept in mind throughout the remainder of this project. However, rather than specifying as to how a criminal pilot can be identified, the questions will be slightly more focussed on whether a correlation can be made between each component. For example, can the flight controller be identified if an investigator was presented with only the Parrot AR Drone 2.0?

Section 2.5.1 has highlighted security vulnerabilities specific to the Parrot AR Drone 2.0. This has provided valuable information with regards to gaining access to this UAV, as well as,
providing a basis for the key areas to look for during the examination stage. It is important to consider how these vulnerabilities could affect a future investigation.

The research related to mobile phone devices, specifically Android devices, has provided a good understanding of how to approach the mobile flight controller component of this investigation. An understanding of the level of volatility of the data stored on a mobile phone device and how to minimise the risk with regards to data acquisition will be key.

In conclusion, the research related to the main components and review of literature presented in this chapter is invaluable in preparing for the remainder of this project. Keeping the key concepts in mind, the second objective of this project can be designed effectively and efficiently.
3. Methodology

This chapter aims to complete the second objective of this study. A practical experiment which is designed to generate data is presented. This chapter will also discuss how each component of this investigation will be acquired and analysed. Section 2.4 has highlighted the importance of UAV forensics. At the time of writing this, a forensic analysis of the Parrot AR Drone 2.0 GPS Edition has not been carried out. Section 3.1 presents the main components included in this investigation. Section 3.2 presents a list of the tools and software that will be used for the acquisition and analysis of the main components. Section 3.3 discusses which methodologies will be applied during the investigation of the Parrot AR Drone 2.0 and the mobile flight controller. Section 3.4 will present a breakdown of how data will be generated for this investigation. This section includes the scenarios that an investigator may come across. Section 3.5 discusses how the tools and software presented in Section 3.2 will be applied to analyse the main components. This also presents what data is expected to be found and where. Section 3.6 presents the ethics related to this investigation. Section 3.7 presents the limitations of this methodology and design.

The methodology for this study will be as follows:

1. The Parrot AR Drone 2.0 and Samsung Galaxy S4 Mini will be used to generate flight data, photo, and video. This data will be examined and analysed under three scenarios. These are discussed in Section 3.4.

2. Section 3.2 provides a brief list of the tools which will be used in the acquisition and analysis of the Parrot AR Drone 2.0, the Samsung Galaxy S4 Mini, the Flight Recorder and the Micro SD card.

3. The Parrot AR Drone 2.0, the Samsung Galaxy S4 Mini, the Flight Recorder, and the Micro SD Card, will be analysed extensively using tools and software. An explanation of how these tools will be used and what data is expected to be extracted is presented in Section 3.5. The aim of this stage is to determine if any data of evidential value can be extracted and how these findings can contribute to a future investigation.
3.1. Main Components

The Parrot AR Drone 2.0 will be the only UAV included in this investigation. This investigation will examine the features of this UAV as discussed in Section 2.3. This will include examining where captured media is stored, and whether flight data, such as GPS is stored.

The Samsung Galaxy S4 Mini running Android version 4.4.2 will be used as the flight controller. The AR Free Flight mobile application is required to pilot the Parrot AR Drone 2.0. AR.FreeFlight 2.4.15 is currently available from Google Play. The features of this application have been discussed in Section 2.3.2. For this investigation, no security features will be applied to the Samsung Galaxy S4 Mini.

The Flight Recorder is a USB type device with 4GB of Flash memory. It acts as both a storage device and a GPS receiver. This investigation will test the features of the Flight Recorder as discussed in Section 2.3.3. This will include testing the ‘Return Home’ and the ‘Control by Map’ feature to determine whether flight data, such as GPS, can be recovered and how this may be relevant. The ‘AR Drone Academy’ will also be examined for flight data.

A formatted 1GB Micro SD memory card will be inserted into the Samsung Galaxy S4 Mini. Examination and analysis of the Micro SD card will determine whether any data is saved or transferred from the Parrot AR Drone 2.0, or from the Samsung Galaxy S4 Mini.

3.2. Tools

The forensic workstation which will be used throughout the acquisition and analysis will provide access to the required software. This includes:

- Kali Linux 2018.1 which will be installed on VM Workstation Pro (Offensive Security, 2018). Kali Linux 2018.1 will be used to gain root access to the OS of the Parrot AR Drone 2.0 via Telnet. A forensic acquisition of the Parrot AR Drone 2.0 will be done using the “dd” command.
- The Tableau Forensic USB Bridge Model T8-R2 write-blocker will be used to prevent write-access to the Flight Recorder.
- Guymager is an imaging tool available on Kali Linux 2018.1. It will be used to take a read-only raw image of the Flight Recorder (Gvoncken, 2016.).
- Cellebrite Phone Detective will be used to identify the Samsung Galaxy S4 Mini and whether any other specific components are needed for acquisition (Cellebrite, n.d.).
- Cellebrite Memory Card Reader device will be used to prevent write-access to the Micro SD Card.
- Cellebrite UFED 4PC will be used to take a physical acquisition of the Samsung Galaxy S4 Mini and the Micro SD memory card (Cellebrite, n.d.).
- Autopsy 4.3.0 will be used as the analysis tool for all of the components included (Carrier, 2017).
- The exiftool is developed by Harvey (2018). It is a free and open-source tool which is simple to use and can be applied to most file types. Exiftool version 10.89 will be used to extract EXIF data from any photo or video files that may be stored.

3.3. Applied Methodology

As discussed in Section 2.1.1, maintaining forensic soundness throughout an investigation is essential. A forensic methodology provides a digital investigator with a systematic approach to a digital investigation. The two main components of this investigation will be the Parrot AR Drone 2.0, and the Samsung Galaxy S4 Mini. It is essential to maintain a systematic approach to the investigation in order to produce and report high quality and professional results.

3.3.1. Parrot AR Drone 2.0 GPS Edition

A forensic methodology which can apply to both main components (UAV and flight controller) has not been presented. However, as discussed in Section 2.4.1, Jain et al. (2017) proposed a forensic methodology which can apply specifically to the examination and analysis of a UAV. Jain et al. have also performed experiments to ensure that this methodology is valid and can be applied generically to any UAV make or model. The investigation of the UAV will, therefore, follow the systematic 12-phase methodology proposed by Jain et al. The diagram presented by Jain et al. demonstrating these 12 phases can be found in Appendix A.

3.3.2. Samsung Galaxy S4 Mini

The challenges and volatility of mobile phone devices make the proposal of a sound forensic methodology difficult. A mobile phone forensic methodology would need to cover every instance of a mobile phone device, for example, how to determine the state of a mobile phone
device without possibly affecting the integrity of the evidence on the mobile phone device or which is the most suitable method of data acquisition for a device which could depend on the make, model and security. Brothers et al. (2014) present what they would consider a suitable methodology for the data acquisition of a mobile phone device. The four core phases include: Preservation, Acquisition, Examination and Analysis, and Reporting. Each phase includes several sub-phases to be followed in order to account for an accurate and forensically sound investigation. Due to the volatile nature of the data stored on a mobile phone device, Brothers et al. recommend enabling “Airplane Mode” on the device. Enabling “Airplane Mode” can mitigate radio communication, thus reducing the risk of the original data being overwritten or modified. This does require some interaction with the device which can pose some risk, however, it is necessary in order to isolate the device and preserve the evidence.

### 3.4. Data Generation

Three scenarios will be created to generate data of evidential value. These scenarios are intended to replicate those which a digital investigator may encounter during an investigation and are inspired by the questions and scenarios discussed in the Section 1.1 as well as the research and literature reviewed in Section 2.4. The three scenarios will be created as follows:

1. Before beginning the experiment, the Parrot AR Drone 2.0 and Samsung Galaxy S4 Mini will be reset to factory settings to ensure that the results of this study are not affected by other variables.
2. The Samsung Galaxy S4 Mini will be connected to the Internet so that AR.FreeFlight 2.4.15 can be installed on the device. Once installed, “Airplane Mode” will be enabled on the device as recommended by Brothers et al. (2014). This approach has been discussed in Section 3.3.2.
3. The first scenario will involve flying the Parrot AR Drone 2.0 in two separate geographical locations. Flight data such as GPS, photo and video will be generated. This scenario is intended to replicate the general use of a UAV without any purposeful modification being made to a component. Each component will be analysed individually to consider the scenario discussed in Section 1.1, i.e. “Can a criminal pilot still be identified if, say, only the UAV or flight controller is found at the scene?” (Marks, 2017).
4. Using the ‘Control by Map’ feature, a flight plan will be designed and executed. Each component will be analysed individually in order to determine whether flight data, such as GPS, is stored. The ‘Return Home’ feature will be tested in order to determine whether the GPS coordinates of the Parrot AR Drone 2.0’s takeoff point are stored as a ‘Return Home’ point. A similar approach was done by Maarse and Sangers (2016). They found artefacts that related to the UAV’s home point and messages of each waypoint being uploaded. This has been discussed in Section 2.4.2.

5. The second scenario will involve the Parrot AR Drone 2.0, the Samsung Galaxy S4 Mini, the Flight Recorder, and the Micro SD card being formatted or reset. Each component will be analysed individually. This scenario is intended to determine if any data of evidential value is still recoverable from any component if, say, a component was damaged or purposefully modified by a ‘criminal’. A similar approach was done by Horsman (2016). He found that data was wiped in an unrecoverable manner. This has been discussed in Section 2.4.3.

6. The final scenario will involve a test of two counter forensic methods. This scenario is inspired by previous studies carried out by Maarse and Sangers (2016) and Horsman (2016). Both studies found that both the time and date stamp, and the GPS data could be manipulated. This has been discussed in Section 2.4.2 and Section 2.4.3. The first method will involve determining whether or not timestamp data can be altered, assuming that any form of timestamp data is stored in the first place. The second method will involve determining whether or not GPS data can be manipulated, assuming that any form of GPS data is stored in the first place.

3.5. Data Analysis

This section discusses how the tools and software listed in Section 3.2 will be used to analyse the main components of this investigation. The key areas of each of the main components are identified as a result of Section 2.4 where UAV forensics has been discussed, and Section 2.5 where the security vulnerabilities of the Parrot AR Drone 2.0 have been discussed.
3.5.1. Parrot AR Drone 2.0 GPS Edition

3.5.1.1. Tools/Software

There is one USB port on the Parrot AR Drone 2.0, which is used for the Flight Recorder. Section 2.5.1 has highlighted a security flaw of this UAV whereby root access to the OS is available via Telnet.

A forensic workstation running Kali Linux 2018.1 on VM Workstation Pro will be used to access the UAV. The built-in Linux “dd” command will be used to copy the partition of the UAV to an external source for analysis. This is a similar approach to that made by Horsman (2016) discussed in Section 2.4.3.

3.5.1.2. Key Areas

A forensic workstation running Autopsy 4.3.0 will be used for analysis. It is unclear which file system the Parrot AR Drone 2.0 uses.

Section 2.5.1 has highlighted a key area where data of evidential value is found. The /data directory will be of interest as this is where files such as “config.ini” reside. This file is likely to contain configuration data related to the Parrot AR Drone 2.0 which may be relevant. Flight data, such as GPS, may also be stored at this location. Media data may also reside here which may contain EXIF data. EXIF data will be examined using exiftool (Harvey, 2018).

3.5.2. Samsung Galaxy S4 Mini Flight Controller

3.5.2.1. Tools/Software

Cellebrite Phone Detective will be used to identify the device and whether a specific cable is required for acquisition.

Cellebrite UFED 4PC will be used to perform a physical acquisition of the Samsung Galaxy S4 Mini. This will provide a bit-by-bit copy of the memory without changing data on the device.

3.5.2.2. Key Areas

A forensic workstation running Autopsy 4.3.0 will be used for analysis. Primary interest will be taken in the /data directory as this is where user installed application data resides. Testing determined the /userdata/data/com.parrot.freeflight directory. Data of evidential value such as flight data is expected, including possibly timestamp data and GPS coordinates of a programmed flight plan.
The /media folder is also of interest as it may contain media related to the UAV. Testing determined the /userdata/media/0/Android/data/com.parrot.freeflight and /userdata/media/0/DCIM/AR.Drone directories. Media files may contain EXIF data. EXIF data will be examined using exiftool (Harvey, 2018).

### 3.5.3. Flight Recorder

#### 3.5.3.1. Tools/Software

A write-blocker and Guymager will be used to acquire a raw image of the Flight Recorder. This will be available on the forensic workstation running Kali Linux 2018.1 on VM Workstation Pro. The Flight Recorder will be acquired after every flight in order to account for any changes made.

#### 3.5.3.2. Key Areas

Media files may be possibly stored on the Flight Recorder. Media files may contain EXIF data. EXIF data will be examined using exiftool (Harvey, 2018). This device also acts as a GPS receiver so a flight log containing GPS data may be recovered.

### 3.5.4. Micro SD Card

#### 3.5.4.1. Tools/Software

Cellebrite’s memory card reader device will be used. A forensic workstation running Cellebrite UFED 4PC will be used to perform a physical acquisition of the Micro SD card. This will provide a bit-by-bit copy of this component.

#### 3.5.4.2. Key Areas

A forensic workstation running Autopsy 4.3.0 will be used for analysis. Media files may be stored on the Micro SD card rather than on the other components. Media files may contain EXIF data. EXIF data will be examined using exiftool (Harvey, 2018). It is not expected that flight data is stored here.
3.6. Ethics

It is vital that the legislation discussed in Section 2.6 is followed. There are further ethical considerations which must be accounted for. As this investigation will involve flying the Parrot AR Drone 2.0 in two separate public locations, it is vital that awareness of the public be taken into account. This includes privacy and consideration for those in the surrounding area. Derbyshire City Council, Derbyshire Constabulary and East Midlands Special Operations Unit have advised the regulations discussed in Section 2.6 and have reiterated that members of the public be made aware of the intent and purpose of this investigation to allow them to leave the area if they wish.

3.7. Limitations

None of the previous literature has identified the file system of the Parrot AR Drone 2.0. Without knowing this, it cannot be determined how data is saved or whether the data is volatile.

The Parrot AR Drone 2.0 has a limited battery life of around 10-15 minutes. The battery takes one hour and a half to fully charge. It may be difficult to generate flight data in two separate locations in the time frame.

As discussed in Section 2.8.1, the nature of data stored on a mobile phone device is volatile. As advised by Brothers et al. (2014), the mobile phone device will be put in Airplane Mode in order to mitigate the risk of data being changed. However, this cannot completely remove the risk of data being changed.

3.8. Conclusions

The design and methodology have been carefully considered and thought through in order to account for, and create, as many real-world scenarios as possible within the time frame of this project. It is essential that the Ethics discussed in Section 3.6, and the current UAV legislation discussed in Section 2.6 are adhered to at all times during this investigation. With this in mind, it is concluded that this is the best approach to achieve the aim of this project.
4. Results and Findings

This chapter aims to complete the third objective of this project whereby the data on the components is examined and the areas of importance are identified. Section 4.1 presents a discussion of some of the limitations found early on in the original design and methodology. As a result, the data generation design was revisited in order to account for these limitations. This revisited data generation design plan is presented in Section 4.2. Section 4.3 presents the findings from the Parrot AR Drone 2.0. Section 4.4 presents the findings from the Samsung Galaxy S4 Mini. Section 4.5 presents the findings from the Flight Recorder. Section 4.6 presents the findings of counter forensic methods from all of the main components.

4.1. Limitations

Section 3.7 discussed the limitations of the methodology and design plan originally proposed. Further limitations which would have impacted the completion of this project were discovered. These limitations are presented in this Section.

A pre-acquisition of the Micro SD card was attempted using Cellebrite’s Memory Card Reader device. The Memory Card Reader device could not communicate with Cellebrite UFED 4PC. The “Continue” button was greyed out, even after following Cellebrite’s instruction. As a result of this, the Micro SD card is removed from the investigation.

A pre-acquisition of the Parrot AR Drone 2.0 revealed many issues. As discussed in Section 3.5.1, the initial plan involved installing Kali Linux 2018.1 on VM Workstation Pro. The forensic workstation would then be connected to the Parrot AR Drone 2.0’s Wi-Fi. Kali Linux 2018.1 would then be used to do an “nmap” scan of the Parrot AR Drone 2.0 and root access would be gained via Telnet if possible. Firstly, the forensic workstation does not have a network card. As a result, the forensic workstation cannot be connected to the Parrot AR Drone 2.0’s Wi-Fi. This limitation is resolved by introducing a second forensic workstation into the investigation. The second forensic workstation is running Windows 10. Kali Linux 2018.1 will instead be installed on a VM Workstation Player 14 on the second forensic workstation. The second forensic workstation can be connected to the Parrot AR Drone 2.0’s Wi-Fi. As a result, this forensic workstation will be used to do an acquisition of the UAV.
The Parrot AR Drone 2.0 has a tiny reset button under the battery. Initially, it was assumed that by pressing this button, the Parrot AR Drone 2.0 would be reset to factory settings, therefore wiping any existing data. Pressing and holding the reset button does reboot the Parrot AR Drone 2.0. This would involve the motors being re-initialised and waiting for the LED’s on the motors to turn from red to green to let the user know that the UAV is ready. However, the existing data on the Parrot AR Drone 2.0 did not change. As this UAV is not brand new, it did have previous flight data on it. Examination of the “config.ini” file revealed data related to the last flight which had taken place. As a solution, a pre-acquisition of the Parrot AR Drone 2.0 will be done. This will provide a capture of the Parrot AR Drone 2.0 in the state before the investigation. Also, the second scenario of the original design as discussed in Section 3.4 involved resetting the Parrot AR Drone 2.0. This would determine whether data can be recovered if any purposeful modification is made. However, the finding that the reset button did not wipe existing data meant that the second scenario of this investigation had to be reconsidered. The revisited design plan is discussed in Section 4.2.

The file system of the Parrot AR Drone 2.0 was initially unknown. As discussed in Section 3.5.1, it was hoped that the “dd” command could be used to make a bit-by-bit copy of the partition of the Parrot AR Drone 2.0 and saved to an external source for analysis. It was found during a pre-acquisition attempt that the Parrot AR Drone 2.0 runs an Unsorted Block Image File System (UBIFS). This presented many challenges in acquiring an image of the Parrot AR Drone 2.0. These are discussed further in Section 4.3.1.

The original design plan discussed in Section 3.4 involved flying the Parrot AR Drone 2.0 in two separate geographical locations. However, the UAV has a very limited flight time of around 10-15 minutes. The battery takes one and a half hours to charge. As a result of this limitation, the scenario’s will be replicated by flying the Parrot AR Drone 2.0 in only one location and generating flight data, photo and video.

4.2. Data Generation Revisited

The data generation method originally discussed in Section 3.4 is revisited in order to consider the limitations discussed in Section 4.1. The main components which will now be involved in this investigation include the Parrot AR Drone 2.0, the Samsung Galaxy S4 Mini as the flight controller, and the Flight Recorder. The scenarios will be created as follows:
1. The Samsung Galaxy S4 Mini will be reset to Factory Settings. As the Parrot AR Drone 2.0 cannot be reset, a pre-acquisition of its current state will be done. This will involve using the second forensic workstation which is running Kali Linux 2018.1 on VM Workstation Player 14. Kali Linux 2018.1 will be used to gain root access to the Parrot AR Drone 2.0 via Telnet.

2. The Samsung Galaxy S4 Mini will be connected to the Internet and the AR.FreeFlight 2.4.15 application will be installed from Google Play. Once installed, “Airplane Mode” will be enabled on the device as recommended by Brothers et al. This has been discussed in Section 3.3.2. Another consideration is also made. As Cellebrite UFED 4PC can only be used on the first forensic workstation, an acquisition of the Samsung Galaxy S4 Mini can only be done after all tests have been carried out. The device will be turned off between flights. This may have some effect on the results.

3. The first scenario will involve flying the Parrot AR Drone 2.0 in one location. Flight data such as GPS, photo and video will be generated. The aim of this scenario is the same as discussed originally. To replicate the general use of a UAV without any purposeful modification being made to a component. Each component will be analysed individually using Autopsy 4.3.0 to consider the scenario discussed in Section 1.1, i.e. “Can a criminal pilot still be identified if, say, only the UAV or flight controller is found at the scene?” (Marks, 2017).

4. Using the ‘Control by Map’ feature, a flight plan will be designed and executed. Each component will be analysed individually in order to determine whether flight data, such as GPS, is stored. The ‘Return Home’ feature will be tested in order to determine whether the GPS coordinates of the Parrot AR Drone 2.0’s takeoff point are stored as a ‘Return Home’ point.

5. The second scenario will involve a flight without the Flight Recorder. This scenario is intended to first determine whether flight data is still stored without this device, assuming that some form of flight data is stored in the first place. This scenario is also intended to determine where media files are stored, assuming that some form of media file is stored in the first place.

6. Two counter forensic methods will be tested. These remain exactly as discussed in Section 3.4. This scenario is inspired by the work of Horsman (2016) and Maarse and Sangers (2016). The first method will involve determining whether or not timestamp data can be altered, assuming that any form of timestamp data is stored in the first place.
The second method will involve determining whether or not GPS data can be manipulated, assuming that any form of GPS data is stored in the first place.

4.3. Parrot AR Drone 2.0 GPS Edition

The Parrot AR Drone 2.0 emits its own Wi-Fi network as “ardrone2_####” where # denotes a random number. This is an unsecured network which is open and visible to anyone in the surrounding area. A forensic workstation running Kali Linux 2018.1 on VM Workstation Player 14 is connected to the Parrot’s Wi-Fi network. Once connected, the command “\texttt{nmap 192.168.1.1}” is run. This reveals that Port 21 and Port 23 are open thus verifying the findings discussed in Section 2.5.1. A Telnet connection provides root access to file system of the Parrot AR Drone 2.0.

Most of the valuable data is stored in the /\texttt{data} directory of the Parrot AR Drone 2.0. Due to complications related to the file system of the Parrot AR Drone 2.0, the only method available to acquire any data is by using the “\texttt{cp –v -a}” command to copy the contents of /\texttt{data} to a USB device mounted to the Parrot AR Drone 2.0. An acquisition of this kind is made after each flight in order to reference the changes made.

4.3.1. Unsorted Block Image File System

The Parrot AR Drone 2.0 is built on an Unsorted Block Image File System (UBIFS). The UBIFS works on top of an Unsorted Block Image (UBI) layer, which in turn, works on top of a Memory Technology Device (MTD) layer.

UBIFS uses a UBI volume to create a reliable and robust file system, adding sub-allocation and garbage collection to create a flash transition layer (Simmonds, 2017). The UBI layer is similar to a volume manager which takes care of bad blocks and wear levelling. UBI provides a reliable view of a flash chip by mapping physical erase blocks (PEB) to logical erase blocks (LEB). UBI keeps count of the number of times each PEB has been erased in the header of the LEB and changes the mapping to ensure that each PEB is erased the same number of times (Simmonds, 2017).

The UBI can access flash memory through the MTD layer. It can divide an MTD partition into a number of UBI volumes. The UBI volume is specified using “\texttt{ubiX\_Y}” or “\texttt{ubiX:NAME}”. X
represents a UBI device number, Y represents a volume number and NAME represents a UBI volume name (Kernel, n.d.).

The MTD provides access to raw flash chips. An MTD can support three operations – read from an offset within an eraseblock, write to an offset in an eraseblock and erase a whole eraseblock (Kernel, n.d.). An MTD partition can be prepared using the \texttt{ubiformat} utility, which preserves the erase counts stored in the headers of the PEB (Simmonds, 2017).

The UBIFS has provided many complications and challenges in acquiring data from the Parrot AR Drone 2.0. Connecting to the Parrot AR Drone 2.0 via Telnet and running the \texttt{“fdisk –l”} command does not reveal a partition in which the \texttt{“dd”} command can be used. Many attempts were made to acquire the full filesystem of the Parrot AR Drone 2.0 in a forensically sound manner, however, these attempts were unsuccessful.

Firstly, the \texttt{“df –h”} command reveals the UBI volumes. The data partition is stored at UBI volume “ubi2:data”. The UBI volumes are stored in \texttt{/dev}. A USB device formatted as FAT32 is mounted at the \texttt{/mnt} directory of the Parrot AR Drone 2.0. An attempt to acquire the UBI volume stored in \texttt{/dev} is done using the following command \texttt{“dd if=/dev/ubi2_1 of=/mnt/ParrotData.dd”}. The USB device is unmounted and inserted into a forensic workstation where it appears that some data has been successfully transferred. However, this file cannot be open or read by any Forensic software such as EnCase or Autopsy as the file system is unrecognised.

A second attempt involved creating a virtual MTD device using \texttt{“nandsim”} to create \texttt{/dev/mtd0} and mounting the “ParrotData.dd” file to this virtual device and analysing the corresponding UBI volume this way, however, this was unsuccessful.

A final option is to connect to the Parrot AR Drone 2.0 through FTP and transfer the files required for analysis this way. In this case, a connection to the Parrot AR Drone 2.0 through FTP is successful, however, any attempt made to copy files close the connection so this did not work. A full list actions taken to attempt to acquire data from the Parrot AR Drone 2.0 can be found in Appendix B.

\subsection*{4.3.2. Flight Data}

The content of each \texttt{/data} is analysed using Autopsy 4.3.0.

The “config.ini” file is stored at \texttt{/data}. This is a configuration log file for the Parrot AR Drone 2.0. This file is modified after each flight in order to update the \texttt{<flying_time>} tag. The
total flight time pre-investigation was `<flying_time = 1747>`, and the total flight time post-investigation is `<flying_time = 3194>`. The `<drone_serial>` tag indicates the serial number of the Parrot AR Drone 2.0. The `<ssid_single_player>` indicates the SSID of the Parrot AR Drone 2.0. Finally, the `<ardrone_name>` indicates the name of the Parrot AR Drone 2.0. These are the most valuable sources of data identified in “config.ini”.

The “syslog.bin” file is stored at `/data`. It contains a log of every event executed by the Parrot AR Drone 2.0. The USB Vendor and USB ProductID are recorded each time a USB device is plugged into the Parrot. Using a USB ID database, the `<USB Vendor: 0x19cf>` and `<USB ProductID: 0x3000>` are identified as “Parrot SA” which suggests that this is the Flight Recorder (The SZ, n.d.).

This file also keeps a detailed log of each session flying time which is used to update the `<flying_time>` of the “config.ini” file.

A folder created at `/data/video/boxes/` is specified as “flight_datestamp_timestamp”. This folder is generated, and the time and date recorded, once a user taps “Take Off” on the pilot view of the AR.FreeFlight 2.4.15 mobile application.

A userbox file is created at `/data/video/boxes/flight_datestamp_timestamp`. This file is specified as “userbox_##########’ where # denotes a random digit. The random digits represent Epoch time. Once converted, this is the same date and time stamp as that of the “flight_datestamp_timestamp” folder.

The userbox file is not in a readable format when viewed through the Kali Linux terminal. When viewed through Autopsy, the userbox file shows information including the serial number, name, and SSID of the Parrot AR Drone 2.0, as well as the make and model of the controller (“.Samsung_GT_I9195:b4eca388873de610”).
A script has been developed by Darklo (2013) and is available on a forum for UAV’s. This script is used to convert the userbox file into a GPS Exchange Format (.GPX) file. This can then be viewed in Google Earth or a Text Editor on Kali Linux. The converted file does not show the data which was previously available in Autopsy. However, the GPS Latitude and Longitude is shown. This is shown in Figure 12.

![Figure 12. Userbox File for Standard Flight Converted to .GPX](image)

When plotted in Google Earth, these coordinates’ display as being in Millennium Wood, Derby, which is accurate to the exact location that this flight was done. If the Parrot AR Drone 2.0 is not given enough time between landing and power down to process the data, the userbox file is not generated. This is shown in Figure 13.
Several configuration files are found at /data/custom.configs/sessions. These are specified as “config.XXXXXXXXX.ini”. The most notable information is the GPS data, which is shown in `<latitude>` and `<longitude>`. The GPS data stored here is reliant on the GPS data as provided by the controller.

The userbox file stored at /data/video/boxes/flight_datestamp_timestamp/ does not store GPS data if the Flight Recorder is not included during flight. This is shown in Figure 14. However, GPS data is still stored at /data/custom.configs/sessions/config.XXXXXXXXX.ini.

```
<?xml version="1.0"?>
<gpx
  version="1.1" creator="userbox_to_gpx.rb"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns="http://www.topografix.com/GPX/1/1"
  xsi:schemaLocation="http://www.topografix.com/GPX/1/1 http://www.topografix.com/GPX/1/1/gpx.xsd">
  <name>flight_20180405_111220</name>
  <trkseg>
  </trkseg>
  </trk>
</gpx>
```

Figure 14. Userbox File Without Flight Recorder.

For this flight, the GPS data shown at /data/custom.configs/sessions/config.XXXXXXXXX.ini is `<latitude = <5.2889876999999998e+01>` and `<longitude = -1.5095419999999999e+00>`. When plotted in Google Earth, these coordinates’ display
as being in Millennium Wood, Derby, which is accurate to the exact location that this flight was done. This is shown in Figure 15.

![Figure 15. Coordinates from /data/custom.configs/sessions/config.XXXXXXXX file in Google Earth.](image)

The suggestion that the GPS data stored in this configuration file is dependent on the GPS information provided by the controller is examined further in Section 4.6 in which Counter Forensic methods have been tested.

Another configuration file is stored at /data/custom.configs/profiles. This file is specified as “config.XXXXXXXX.ini”. This configuration file contains information related to the flight controller. The <profile_desc> tag shows the make, model and serial number of the flight controller. In this case, this appears as <profile_desc = .Samsung_GT_I9195:b4eca388873de610>.

### 4.3.3. Media

Initial examination suggests that no media files are stored on the Parrot AR Drone 2.0. In the first standard flight, photo files are stored on the flight controller and video files are stored on the Flight Recorder.
Without the Flight Recorder, however, photo files are stored on the Parrot AR Drone 2.0 at /data/video/boxes/flight_datestamp_timestamp, along with the created userbox file. Each photo stored is specified as “picture_datestamp_timestamp”. These are stored as Joint Photographic Expert Group (.jpg) files by default.

Photo files are analysed using exiftool (Harvey, 2018). The most important information is <Make> which displays the Make of the device used to take the photo (<Make: Parrot AR.Drone>), the <Camera Model Name> displays the model of the camera used to take the photo (<Camera Model Name: Samsung SOC1040>), the <Date/Time Original> and <Create Date> display the time and date that the photo is taken. Finally, the <GPS Altitude>, <GPS Latitude>, <GPS Longitude> and <GPS Position> is supposed to display the GPS information related to each photo. However, only the <GPS Latitude> value is given. No value is given for <GPS Longitude> for any of the photo files stored.

An online tool developed by Friedl (2006) is used to verify the validity of the EXIF data and determine whether any GPS information can be extracted. The <Camera> displays the make and model of the camera used to take the photo (<Camera: Parrot AR.Drone Samsung SOC1040>). The <Date> displays the date and time that the photo was taken. The <Location> only provides a Latitude value. This value varies slightly for each photo. The Longitude value is not given for any photo. An accurate point of location, therefore, cannot be determined.

At no time is a video file stored on the Parrot AR Drone 2.0.

### 4.4. Samsung Galaxy S4 Mini Flight Controller.

An acquisition of the Samsung Galaxy S4 Mini is taken using Cellebrite UFED 4PC. This acquisition is done only after all of the flights are complete. Forensic soundness was maintained by putting the Samsung Galaxy S4 Mini in Airplane Mode as suggested by Brothers et al. (2014).

#### 4.4.1. Flight Data

Upon installing the AR.FreeFlight 2.4.15 mobile application, a new folder is created at /userdata/data/ is specified as “com.parrot.freeflight”. This folder contains several .xml files specified as ”MAC address of controller_datetimestamp”. Each file appears to correlate with
when each flight was performed. The `<info key="FLIGHT_TIMESTAMP_FLIGHT_STARTED>` and `<info key="FLIGHT_TIMESTAMP_FLIGHT_ENDED>` contain a date and time stamp for when a flight started and stopped. The `<info key="APPLICATION_SESSION_DEVICE_MAC_ADRESS>` tag indicates the MAC address of the controller. The `<info key="FLIGHT_DRONE_SERIAL>` tag indicates the serial number of the Parrot AR Drone 2.0. The `<info key="APPLICATION_SESSION_DEVICE_MODEL>` indicates the make and model of the flight controller and `<info key="APPLICATION_SESSION_DEVICE_OS_VERSION>` indicates the OS version. The `<info key="APPLICATION_SESSION_TIMESTAMP_APPSTARTED">` and `<info key="APPLICATION_SESSION_TIMESTAMP_APPLICATION_CLOSED>` indicate when the FreeFlight application was opened and closed.

Several folders are stored at `/userdata/media/0/DCIM/AR.Drone/`. Each folder is specified as “flight_datestamp_timestamp”. Each folder contains a userbox file, exactly like those discussed in Section 4.3.2. When viewed through Autopsy, the userbox file displays key information including the name, SSID, and the serial number of the Parrot AR Drone 2.0 and the make and model of the controller.

The script developed by Darklo (2013) is used to convert the userbox file into a .GPX file. The converted file does not show the data which was previously available in Autopsy. However, the GPS Latitude and Longitude is shown. There is no difference between the userbox file which is stored on the Parrot AR Drone 2.0 and the userbox file which is stored on the controller. Since the “flight_datestamp_timestamp” folder stored on the Parrot AR Drone 2.0 is overwritten with each new flight, the folder and its contents are transferred to the flight controller.

### 4.4.2. Media

Photo files are stored on the controller at `/userdata/media/0/DCIM/AR.Drone`. These are specified as “picture_datestamp_timestamp”. These are stored as .jpg by default. If the Flight Recorder is not included during flight, video files will store on the flight controller. In this case, the video file is specified as “video_datestamp_timestamp”.
Photo files are examined using exiftool (Harvey, 2018). The <Make> tag reveals that the photo is taken using a “<Make: Parrot AR.Drone>”. The <Camera Model Name> provides the model of the camera used to take the photo, “<Camera Model Name: Samsung SOC1040>”. The <Date/Time Original> and <Create Date> provides the date and time that the photo was taken. There are four tags which are supposed to include GPS information <GPS Altitude>, <GPS Latitude>, <GPS Longitude> and <GPS Position>, however, only the <GPS Latitude> value is given. No value is given for <GPS Longitude> for any of the photo files stored.

The photo files are also analysed using an online EXIF tool to verify whether any EXIF data such as GPS can be recovered (Friedl, 2006). The <Camera> tag provides the make and model of the camera used to take the photo, i.e. “<Camera: Parrot AR.Drone Samsung SOC1040>”. The <Date/Time Original> and <Create Date> provides the date and time that the photo was taken. The <Location> only provides a Latitude value. This value varies slightly for each photo. The Longitude value is not given for any photo. An accurate point of location, therefore, cannot be determined.

The video file is examined using exiftool (Harvey, 2018). The <Create Date> displays the date and time of when the video was taken. The <Vendor ID > provides the Make of the camera used to take the video, in this case <Vendor ID: Parrot AR.Drone> and <Software Version> provides further information <Software Version: AR.Drone 2.0>.

4.4.3. Other Relevant Data

The file “wpa_supplicant.conf” is stored at /userdata/misc/wifi/. This file contains information about which Wi-Fi hotspots the flight controller has connected to. Using the information provided here, an investigator can verify that a connection between the controller and the Parrot AR Drone 2.0 has been made. This is shown as “ssid = “ardrone2_XXXXXX””.

The file “mac.info” is stored at /efs/wifi/. This shows the MAC address of the controller. This will prevent an investigator from viewing the MAC address of the device manually using the device settings, therefore mitigating the risk of data being changed.
Finally, the file “serial_no” is stored at /efs/FactoryApp/. This contains the serial number of the controller. This will prevent an investigator from checking this manually using the device settings, therefore mitigating the risk of data being changed.

4.5. Flight Recorder

The Flight Recorder is a small USB type device with 4GB of flash memory which is used for storing media and acts as a GPS receiver. A raw image acquisition of the Flight Recorder was taken after each flight using Guymager running on Kali Linux. The raw image generated is then analysed in Autopsy 4.3.0.

4.5.1. Flight Data

Flight data such as GPS information is not available from video files. Various tools were used to examine whether any GPS information could be extracted which were unsuccessful.

4.5.2. Media

Folders are automatically created on the Flight Recorder. These folders are specified as “media_datestamp_timestamp”. The videos taken are stored in the folder corresponding to its flight. These are specified as “video_datestamp_timestamp”. These are stored as Moving Picture Experts Group (.mp4) by default.

The video files are examined using exiftool (Harvey, 2018). The <Create Date> indicates the date and time of when the video was taken. The <Vendor ID> indicates the make of the camera used to take the video, in this case <Vendor ID: Parrot AR.Drone> and <Software Version> provides further information (<Software Version: AR.Drone 2.0>). At no time is a photo file stored on the Flight Recorder.

4.6. Counter Forensic Methods

4.6.1. Parrot AR Drone 2.0 – Time and Date Manipulation

The time and date of the controller are intentionally changed to test whether this manipulation is reflected on any of the data stored on any component included in this investigation.
The Parrot AR Drone 2.0 relies on the controller to provide an accurate time and date. The incorrect time and date is reflected on the “flight_datestamp_timestamp” folder at `/data/video/boxes/`.

The userbox file stored at `/data/video/boxes/flight_datestamp_timestamp` stores the time and date of a flight as well as GPS data. The manipulated time and date is reflected on the userbox file. This is shown in Figure 16 where the manipulated time and date is identified in `<time>`.

![Figure 16. Userbox File Showing Manipulated Time and Date.](image)

Photo files are stored on the Parrot AR Drone 2.0 at `/data/video/boxes/flight_datestamp_timestamp/`. These files are specified as “picture_datestamp_timestamp.jpg”. The manipulated time and date is reflected in the file name. The manipulated time and date has also been embedded into the EXIF data of the photo files also.

### 4.6.2. Samsung Galaxy S4 Mini – Time and Date Manipulation

As discussed in Section 4.4.1, flight data is stored on the flight controller in .xml files at `/userdata/data/com.parrot.freeflight/`. These .xml files are specified as “MAC address of controller_datestamptimestamp”. The incorrect time and date is reflected on the .xml file name.
corresponding to this flight. However, this is also the only .xml file of several, which has not stored the MAC address of the flight controller in either the file name or within the body of the file.

As discussed in Section 4.4.2 photo files are stored on the flight controller at /userdata/media/0/DCIM/AR.Drone/. These files are specified as “picture_datestamp_timestamp”. The incorrect date and time is reflected on the photo file name. The incorrect date and time stamp is also embedded in the EXIF data of the photo file. This is tested viewing through both Autopsy and exiftool (Harvey, 2018).

### 4.6.3. Flight Recorder – Time and Date Manipulation

As discussed in Section 4.5.2, video files are stored on the Flight Recorder in folders specified as “media_datestamp_timestamp”. The incorrect time and date is reflected on the folder corresponding with this flight. Video files stored within this media folder are specified as “video_datestamp_timestamp”. The incorrect time and date is reflected on the file name also. Using exiftool (Harvey, 2018), the `<Create Date>` displays the actual and correct time and date that the video was taken. This is the only file across all components that displays the correct time and date, despite the incorrect time and date being reflected in the file name. This is shown in Figure 17.

![Figure 17. EXIF Data of Video File showing Correct Time and Date.](image)
4.6.4. Parrot AR Drone 2.0 – GPS Manipulation

As discussed in Section 4.3.2, without the Flight Recorder GPS information is not recorded in the userbox file, but GPS information is identified at /data/custom.configs/sessions/configXXXXXXX.ini. This suggests that the GPS data identified in /data/custom.configs/sessions/configXXXXXXX.ini is provided by the flight controller. A test to conclude this suggestion is made.

A mobile application which allows users to spoof their location is installed on the controller. A random location is selected. The coordinates for which are 43.746013, -79.310140. Google Maps reveals this location as Toronto, Canada. This is shown in Figure 18. The Flight Recorder was also included and active in this test.

Figure 18. GPS Manipulated Coordinates in Toronto, Canada.
Firstly, an examination of the userbox file generated and stored on the Parrot AR Drone 2.0 at /data/video/boxes/flight_diestamp_timestamp reveals the correct GPS coordinates for where this flight test took place. This is shown in Figure 19.

![Google Earth GPS Data Displayed in Userbox File](image.png)

**Figure 19. GPS Data stored in Userbox File for GPS Manipulated Flight.**

When plotted in Google Earth, these coordinates’ display as being in Millennium Wood, Derby. This is shown Figure 20. This is accurate to the real location that this flight was carried out.

![Google Earth GPS Data Plotted](image.png)

**Figure 20. GPS Data stored in Userbox File as Plotted in Google Earth.**
The configuration file generated for the same flight and stored on the Parrot AR Drone 2.0 at /data/custom.configs/sessions/config.XXXXXXXX.ini however, reveal the GPS coordinates \(<\text{latitude} = 4.3746014000000002\text{e+01}>\text{ and } <\text{longitude} = -7.931014299999997e+01>\). When plotted in Google Earth, these coordinates display as Toronto, Canada. This is very accurate to the exact spoofed location. This is shown in Figure 21.

This therefore suggests that the GPS information found /data/custom.configs/sessions/config.XXXXXXXX.ini relies on the GPS information provided by the flight controller.

As discussed in Section 4.3.3, photo files are sometimes stored on the internal storage of the Parrot AR Drone 2.0 at /data/video/boxes/flight_datestamp_timestamp. Each photo taken during the GPS Spoof flight is examined using the online EXIF tool (Friedl, 2006). The GPS Latitude value is provided as \(<\text{GPS Latitude: } 42.745833\text{. The GPS Longitude value is not given. Without both values, it is difficult to determine the location that this photo was taken. However, the } <\text{GPS Latitude} >\text{ value suggests that the spoofed GPS Latitude has been embedded into the photo.}\\n\\n\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{image}
\caption{Manipulated GPS Data Stored in "config.XXXXXXXX.ini" as Plotted in Google Earth.}
\end{figure}
4.6.5. **Samsung Galaxy S4 Mini – GPS Manipulation**

The usertbox file which should store on the flight controller at /
userdata/media/0/DCIM/AR.Drone/flight_datestamp_timestamp is not available. This file will probably transfer from the Parrot AR Drone 2.0 once a new flight is carried out. The usertbox is exactly the same so will show the same GPS data as shown in Figure 19.

The photo files corresponding to this test are analysed using an online EXIF tool (Friedl, 2006). The GPS Latitude value is provided as <GPS Latitude: 43.745833>. The GPS Longitude value is not given. Without both values, it is difficult to determine the location that this photo was taken. However, the <GPS Latitude> value suggests that the spoofed GPS Latitude has been embedded in EXIF data of the photo.

4.7. **Conclusion**

This chapter has completed the third objective of this project whereby the areas of importance have been identified. The expectation of what may be found in the key area of each component as discussed in Section 3.5 was mostly correct.

From these findings, it is clear that each component included in this investigation has its own particular relevance. For example, the Samsung Galaxy S4 Mini has limited GPS data, which is made up with the Parrot AR Drone 2.0. The Parrot AR Drone 2.0 has limited storage for media, which is made up with the Samsung Galaxy S4 Mini and the Flight Recorder.

The finding that the Parrot AR Drone 2.0 will store two forms of GPS data in any one flight is very interesting, particularly since one set of GPS data can be manipulated using a mobile application. Also, finding that the time and date can be manipulated is very important. This will be discussed in the next chapter.

It was expected that the “Control by Map” and “Return Home” features would be tested. Unfortunately, due to certain circumstances, this was not possible. The installation program required for these features to work would not install. As a result, neither feature could be tested as hoped.
5. Discussion and Analysis

This chapter aims to complete the final objective of this study in which the findings of the investigation are discussed and the relevance of these findings. The aim of Chapter 3 originally was to replicate three scenarios based on the questions asked in Section 1.1. “How can a criminal pilot be identified if only the UAV is found at the crime scene? Or if only fragments from a wreckage are found? Or when only the flight controller or UAV is found?” (Marks, 2017). The questions were expanded further for this investigation to consider more than just identifying a criminal pilot.

Section 5.1 presents three tables which have identified the key findings across the Parrot AR Drone 2.0, the Samsung Galaxy S4 Mini as the flight controller, and the Flight Recorder. Section 5.2.1 discusses the relevance of the findings presented in this paper compared to previous studies. Section 5.2.2 discusses the relevance of the findings presented in this paper for law enforcement. Section 5.3.1 will discuss what data identified from the Parrot AR Drone 2.0 can be used to identify the Samsung Galaxy S4 Mini flight controller. Section 5.3.2 will discuss what data from the Samsung Galaxy S4 Mini is identified which will identify the Parrot AR Drone 2.0. Section 5.3.3 will discuss what data is identified on the Flight Recorder can be used to identify either the Parrot AR Drone 2.0 or the Samsung Galaxy S4 Mini flight controller. Section 5.4 will present an analysis of the results obtained in the Time and Date manipulation, and the GPS manipulation.

5.1. Key Findings

This section summarises the key findings of the main components identified in Chapter 4. The file path, file name, and what data is identified are presented as tables below. Table 1 summarises the findings identified on the Parrot AR Drone 2.0. Table 2 summarises the findings identified on the Samsung Galaxy S4 Mini flight controller. Finally, Table 3 summarises the findings identified on the Flight Recorder.
### Parrot AR Drone 2.0 – GPS Edition

<table>
<thead>
<tr>
<th>File Path</th>
<th>File Name</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>/data</td>
<td>config.ini</td>
<td>Total flight time, UAV serial number, SSID of UAV, name of UAV</td>
</tr>
<tr>
<td>/data</td>
<td>syslog.bin</td>
<td>Detailed log of actions and commands related to UAV, USB Vendor, USB Product ID.</td>
</tr>
<tr>
<td>/data/video/boxes/flight_datestamp_timestamp/</td>
<td>userbox_################</td>
<td>GPS, time and date stamp</td>
</tr>
<tr>
<td>/data/video/boxes/flight_datestamp_timestamp/</td>
<td>picture_datestamp_timestamp.jpg</td>
<td>EXIF data - Make and model of camera, time and date stamp</td>
</tr>
<tr>
<td>/data/custom.configs/profiles/</td>
<td>config.XXXXXXXXX.ini</td>
<td>Make, model and serial number of controller</td>
</tr>
<tr>
<td>/data/custom.configs/sessions/</td>
<td>config.XXXXXXXXX.ini</td>
<td>GPS from the flight controller.</td>
</tr>
</tbody>
</table>

Table 1: Summary of findings identified on Parrot AR Drone 2.0.

### Samsung Galaxy S4 Mini – Flight Controller

<table>
<thead>
<tr>
<th>File Path</th>
<th>File Name</th>
<th>Relevance</th>
</tr>
</thead>
<tbody>
<tr>
<td>/userdata/data/com.parrot.freeflight</td>
<td>MAC address_datetimestamp.xml</td>
<td>Flight start and stop time and date stamp, MAC of controller, serial number of UAV, make and model of controller.</td>
</tr>
<tr>
<td>/userdata/media/0/DCIM/</td>
<td>userbox_################</td>
<td>GPS, date and time stamp</td>
</tr>
<tr>
<td>AR.Drone/flight_datestamp_timestamp</td>
<td>EXIF data – Make and model of camera, time and date stamp.</td>
<td></td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>/userdata/media/0/DCIM/AR.Drone/</td>
<td>picture_datestamp_timestamp.jpg</td>
<td></td>
</tr>
<tr>
<td>video_datestamp_timestamp.mp4</td>
<td>EXIF date - Original time and date stamp, make and software version of camera.</td>
<td></td>
</tr>
<tr>
<td>/userdata/misc/wifi/</td>
<td>wpa_supplicant.conf</td>
<td></td>
</tr>
<tr>
<td>/efs/wifi/</td>
<td>mac.info</td>
<td></td>
</tr>
<tr>
<td>/efs/FactoryApp/</td>
<td>serial_no</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Serial number of controller.</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Summary of findings identified on Samsung Galaxy S4 Mini.

Table 3: Summary of findings identified on Flight Recorder.

| Flight Recorder |
|-----------------|-----------------|-----------------|
| File Path       | File Name       | Relevance       |
| USB DRIVE/media_datestamp_timestamp | video_datestamp_timestamp.mp4 | EXIF data – Original time and date stamp, make and software version of the camera. |

5.2. Relevance

5.2.1. Compared to Previous Literature

The results and findings presented in Chapter 4 are similar to those discussed in Section 2.4. Flight data is recoverable from the Parrot AR Drone 2.0. The integrity of the data stored can be affected by turning the UAV on. This is similar to the findings presented by Clarke (2017) and Horsman (2016) whereby a new file is created every time the UAV is turned on. Previous flight data was not recoverable. Media files are identified on all three of the components included in this investigation. EXIF data is also retrievable. Maarse and Sangers (2016) and Horsman (2016) were able to identify the longitude and latitude coordinates in their investigations. However, in this case, only the Latitude value is ever given for the media files. It is not clear
why this is the case. Furthermore, other relevant files were identified on the flight controller, including the SSID of any Wi-Fi network that the flight controller had connected to (Refer to Table 2). This form of data was not presented in any other literature but can be used to indicate that a flight controller has connected to the UAV.

With regards to counter forensic methods, the findings presented in this investigation are similar to those presented by Maarse and Sangers (2016) and Horsman (2016). Both the time and date, and the GPS data can be manipulated using simple methods. This is a problem for investigators, as there is no valid method of determining that the time and date or the GPS data identified on either a UAV or its flight controller is reliable.

In the case that an investigator is presented with all of the main components, there is a lot of important data that can be identified across all three components. However, there may be a case where an investigator is presented with only one or two of these components.

5.2.2. For Law Enforcement

As discussed in Section 1.1, UAV’s can be used for illegal offences and activity. These include spying, illegal streaming and smuggling. The findings presented in Chapter 4 can help law enforcement build a case against a potential criminal.

As discussed in Section 4.3.2 there are two sets of GPS data stored on the Parrot AR Drone 2.0. An investigator can use this data to prove that a criminal was in a certain place. Of the two sets of GPS data stored on the Parrot AR Drone 2.0, only the GPS data stored in the userbox file is correct. The second set of GPS data can be manipulated and therefore should not be considered reliable. Without a current UAV registration process in place, there is no way to prove ownership of the UAV (Refer to Section 2.6.3).

Media data is mostly found on the flight controller and Flight Recorder. However, there is no way to identify who took a photo or video. There is also no way to determine the exact location that a photo or video was taken. For photo files, only the \textit{Latitude} value is ever given in EXIF data. For video, no GPS data was recovered. It is also found that the time and date that a photo is taken can be manipulated (Refer to Section 4.6). Therefore, the time and date of a photo should not be entirely relied. The manipulated time and date is shown in the filename of a video, however, EXIF data reveals the correct time and date (Refer to Section 4.6.3). An investigator should therefore check the EXIF data of video files in order to verify the time and date.
5.3. Correlation Between Main Components

The aim of this section is to discuss whether the data identified on any of the components in as discussed in Chapter 4 can be used to aid an investigation if, say, an investigator was presented with only one component to examine and analyse. For example, can the flight controller and/or the Flight Recorder be identified if the investigator was only presented with the Parrot AR Drone 2.0. This is a general approach to the question “How can a criminal pilot be identified if only the UAV is found at the crime scene? Or if only fragments from a wreckage are found? Or when only the flight controller or UAV is found?” (Marks, 2017), asked in Section 1.1. This analysis will be done on all of the components involved in this investigation.

5.3.1. Parrot AR Drone 2.0 GPS Edition

This analysis will determine whether the flight controller, in this case the Samsung Galaxy S4 Mini, and/or the Flight Recorder can be identified using the data identified on the Parrot AR Drone 2.0. The findings of the Parrot AR Drone 2.0 are discussed in Section 4.3. The /data/custom.configs/profiles/config.XXXXXXXX.ini is the only file identified on the Parrot AR Drone 2.0 which can identify the Samsung Galaxy S4 Mini flight controller. This file contains data related to the make, model, and the serial number of the flight controller. The /data/custom.configs/sessions/config.XXXXXXXX.ini contains flight data, specifically GPS data. The GPS data stored in this file is reliant on the GPS data provided by the flight controller. However, examination revealed that this GPS data can be manipulated, so it is not a reliable source of data. This is discussed in Section 4.6.4.

In the absence of the flight controller, these are the only two sources of data which are identified from the Parrot AR Drone 2.0 which can associate that the UAV and Samsung Galaxy S4 Mini have been connected. Most of the data acquired from the /data directory is volatile. This includes both “config.XXXXXXXX.ini” files. Once overwritten with new flight data, the previous flight data was not recoverable.

The “syslog.bin” file can be used to determine which devices have been physically connected to the Parrot AR Drone 2.0. This is the only file which can identify the presence of the Flight Recorder. The USB Vendor and USB ProductID for the Flight Recorder are identified in Section 4.3.2. In a case which may involve possible criminal activity, an investigator could identify another USB device with the USB Vendor and USB ProductID tags, which may not have been considered relevant in the investigation before. Other than identifying the USB
Vendor and USB ProductID, no other detail of the Flight Recorder is identified on the Parrot AR Drone 2.0.

### 5.3.2. Samsung Galaxy S4 Mini Flight Controller

This analysis will determine whether the Parrot AR Drone 2.0 and/or Flight Recorder can be identified using only the flight controller. There is a wealth of data identified on the Samsung Galaxy S4 Mini flight controller which can be used to associate that a Parrot AR Drone 2.0 have been connected.

The time and date of when a flight started and stopped was identified at `/userdata/data/com.parrot.freeflight/MAC address of controller_dateandtimestamp.xml`. Firstly, this can prove that this mobile phone device has been used as a flight controller. This file also contains the serial number of the Parrot AR Drone 2.0. Every UAV will have a unique serial number. A Google search reveals that there is no current central database which can be used to lookup the details of a UAV using the serial number. This file alone does not show any further detail such as the make or model of the UAV, so the serial number could be useless unless further detail can be identified. Even so, there are probably hundreds to thousands of Parrot AR Drone 2.0’s around the world. It would be difficult to search every one of these to identify the correct Parrot AR Drone 2.0. The use of the serial number would only be useful if the investigator had an idea of which UAVs were possibly involved in a crime. The serial number could then be used to identify the exact one. Alternatively, if a UAV registration process was currently in place, then the serial number could be used to identify the legitimate owner of the UAV.

The “wpa_supplicant.conf” file is identified in Section 4.4.3. This file contains a list of Wi-Fi connections made by the flight controller. In this case the Parrot AR Drone 2.0 is identified as “ssid = ardrone2_XXXX”. This verifies that the Samsung Galaxy S4 Mini has connected to the Parrot AR Drone 2.0, however, this alone does not confirm anything more. The flight controller can still connect to the Wi-Fi emitted by the Parrot AR Drone 2.0. However, none of the features, such as taking photo and video, can be used without the AR.FreeFlight 2.4.15 application. If the “com.parrot.freeflight” folder is not identified at `/userdata/data/`, it can only be assumed that the flight controller connected to the Parrot AR Drone 2.0, but did not use any of its features.
Media files are identified on the flight controller in Section 4.5.2. Firstly, if an investigator has identified media files on the device, this can verify that the flight controller has connected to, and communicated with, the Parrot AR Drone 2.0. The photo and video feature cannot be used without the AR.FreeFlight 2.4.15 application. Also, photo and video files cannot be taken or saved without a connection between the Parrot AR Drone 2.0 and the flight controller. The EXIF data of these files can be used to identify the make and model of the UAV. In this case, the make is shown as <Make: Parrot AR Drone 2.0> and model is identified as <Model: Samsung SOC1040>.

GPS data was identified on at /userdata/media/0/DCIM/AR.Drone/flight_datestamp_timestamp/. The “flight_datestamp_timestamp” folder and its contents are transferred from the Parrot AR Drone 2.0 to flight controller each time a new flight is carried out by a user. Unlike the “flight_datestamp_timestamp” folder stored on the Parrot AR Drone 2.0, this data is not as volatile and is still available after the flight controller has been restarted. Identifying the “flight_datestamp_timestamp” on the Samsung Galaxy S4 Mini verifies that the flight controller has connected to, and communicated with, the Parrot AR Drone 2.0 as this folder cannot be transferred otherwise.

Examination of this file using Autopsy revealed data specified to the Parrot AR Drone 2.0, including the name, SSID, and the serial number of the UAV. GPS data is only readable once converted, though the converted file does not show the data which was previously available through Autopsy. Converting the file does not affect the original as a copy of the file is made and saved to the location specified by the user. The GPS data stored in this userbox file is reliant on the GPS receiver of the Flight Recorder. In the absence of the Flight Recorder, the userbox file stored on the Parrot AR Drone 2.0 does not store GPS data. Therefore, when the userbox file is transferred to the flight controller, the GPS data is not available. This is identified in Section 4.3.2 and shown in Figure 14. During this investigation, it was also found that if the Parrot AR Drone 2.0 is not given enough time between landing and power off, then the userbox file is not generated. In such case, no userbox file is transferred to the flight controller at all. The userbox file is also identified as being the only source of GPS flight data available on the flight controller.

In the instance that only the flight controller is presented in an investigation, there is a wealth of relevant and valuable data which can be identified and associated with the Parrot AR Drone 2.0. Many of the files and folders identified cannot be stored on the flight controller unless the
flight controller and Parrot AR Drone 2.0 have connected, and communicated, with each other. Other than the userbox file, there is no other source of data on the flight controller which can identify the presence of the Flight Recorder. If the investigator can identify the userbox file, and it contains GPS data then it can be assumed that the Flight Recorder was present during a flight.

5.3.3. Flight Recorder

The data stored on the Flight Recorder is very limited. At no time is a photo file, or flight data, stored on the Flight Recorder. The only source of data identified on the Flight Recorder is in the EXIF data of the video files. The tag `<Make: Parrot AR Drone>` reveals that a Parrot AR Drone device took the video, however, the EXIF data does not reveal any further detail. Detail related to the flight controller was not identified on the Flight Recorder.

5.4. Counter Forensic Methods

5.4.1. Parrot AR Drone 2.0 – Time and Date Manipulation

It is concluded that the Parrot AR Drone 2.0 relies on the flight controller to provide an accurate time and date. Examination revealed that manipulation of the flight controller’s time and date is reflected on all data stored on the Parrot AR Drone 2.0. The manipulated time and date is reflected on the “flight_datestamp_timestamp” folder stored at `/data/video/boxes/`. The userbox file containing flight data is stored in this folder. As well as GPS data, the userbox file stores a time and date stamp of the flight. Examination revealed that the manipulated time and date is reflected in the content of the userbox file also. This is shown in Figure 16. Photo files will store on the Parrot AR Drone 2.0 in the absence of the Flight Recorder. Photo files are specified as “picture_datestamp_timestamp.jpg”. The manipulated time and date are reflected in the file name of the photo. It is also reflected in the EXIF data of the photo files.

These findings indicate that the time and date stamp cannot be fully relied on for accuracy or validity. The ability to change the time and date of a mobile phone device is not a difficult task. A ‘criminal’ could manipulate the time and date of the flight controller and make many claims, such as they did not own the UAV at the time or date an incident took place, or they may claim that they were away during the time or date an incident took place. There was no way to prove the accuracy and validity of the time and date during this investigation. In the instance that an
investigator is presented with the Parrot AR Drone 2.0, it is difficult to prove the accuracy of
the time and date stamp reflected in or on the files.

5.4.2. Samsung Galaxy S4 Mini – Time and Date Manipulation

Flight data was identified as being stored on the flight controller at /userdata/data/com.parrot.freeflight in .xml files specified as “MAC address of controller_datestamp_timestamp.xml”. Examination revealed that the manipulated time and date is reflected on the file name and within the body of the .xml file.

Examination revealed that the manipulated time and date is reflected on the file name for photo files. It is also reflected in the EXIF data of the photo files.

It was identified that in the absence of the Flight Recorder, video files were stored on the flight controller at /userdata/media/0/DCIM/AR.Drone/. These files are specified as “video_datestamp_timestamp”. The manipulated time and date is reflected on the file name.

These findings indicate that the time and date reflected in or on files or folders cannot be fully relied on for accuracy or validity. There is no way to prove the accuracy and validity of the time and date for other files stored on the Samsung Galaxy S4 Mini flight controller. In the instance that an investigator is presented with the flight controller, it is difficult to prove the accuracy of the time and date as it is reflected in or on the files or folders.

5.4.3. Flight Recorder – Time and Date Manipulation

Folders created on the Flight Recorder are specified as “media_datestamp_timestamp”. The manipulated time and date is reflected on the folder name. Video files are stored within the folder which corresponds with when it was taken. These files are specified as “video_datestamp_timestamp.mp4”. The manipulated time and date is also reflected on the video file name. However, analysis of the video file using exiftool (Harvey, 2018) reveal the actual time and date that the video was taken (Refer to Figure 17). This is the only file across all components that shows the correct time and date, despite the manipulated time and date being reflected in the file name. This suggests that the Flight Recorder, or the video feature, possibly relies on another source or component for an accurate time and date. It is not concluded that the accurate time and date will always show in the EXIF data of a video file which has been
purposely manipulated. Therefore, in the instance that an investigator is presented with only the Flight Recorder, they should not fully rely on the accuracy or validity of the time or date stamp until they have analysed the file using an EXIF tool.

5.4.4. Parrot AR Drone 2.0 – GPS Manipulation

The GPS data recorded in the userbox file is reliant on the data provided by the Flight Recorder. If the Flight Recorder is present during a flight, the GPS data recorded in the userbox file is correct and is not affected by the manipulated GPS of the flight controller (Refer to Figure 19). In the absence of the userbox file, an investigator must rely on the GPS data stored on the Parrot AR Drone 2.0 at /data/custom.configs/sessions/configXXXXXXXX.ini. It is concluded that the GPS data shown in this file relies on GPS being accurately provided by the flight controller. The manipulated GPS is very accurately reflected in this file. In the instance that an investigator is presented with only the Parrot AR Drone 2.0, they may straight away consider examining the userbox file for flight data. However, as demonstrated in this investigation, there can be instances where the userbox file is not created, or that the Flight Recorder was not present during a flight and therefore no GPS data is stored in the userbox file. In this case, the only GPS data available for an investigator to examine is /data/custom.configs/sessions/config.XXXXXXXX.ini. However, the findings of this investigation as discussed in Section 4.6.4 conclude that this GPS data can be manipulated quite easily.

Examination of the EXIF data for the photo files stored on the Parrot AR Drone 2.0 revealed that the <Latitude> value of the manipulated GPS had been embedded into the photo. This is discussed in Section 4.3.3. Without both the <Latitude> and the <Longitude> value, an investigator would not be able to rely on the GPS EXIF data of a file anyway. However, this does suggest that the manipulated GPS can be embedded into a photo and therefore, should not be relied on for complete accuracy if there is an instance where both values are provided.

5.4.5. Flight Controller – GPS Manipulation

The userbox file for this flight should have transferred from the Parrot AR Drone 2.0 to the flight controller. This file should be located at /userdata/media/0/DCIM/AR.Drone/flight_datestamp_timestamp. The userbox file for this specific flight could not be identified. It is possible that this userbox file will transfer to the
flight controller the next time that a flight is carried out. However, as the userbox file is transferred from the Parrot AR Drone 2.0, it is expected that the correct GPS data would be shown as this is the file which relies on the Flight Recorder to provide accurate GPS data.

In the instance that an investigator is presented with only the flight controller, the userbox file is the only source of GPS data identified. The GPS data presented in this file is correct and accurate. However, there can be a chance that the userbox file is not present on the flight controller. If the Flight Recorder is not present during a flight, then the userbox file which is transferred to the flight controller will not provide any GPS data (Refer to Figure 14). Also, as demonstrated in this investigation, the userbox file is only transferred from the Parrot AR Drone 2.0 to the flight controller the moment that a new flight is carried out. This suggests that if a ‘criminal’ has not carried out a new flight or is using a different mobile phone device as a flight controller, then the userbox file will not be transferred to the flight controller in the investigation.

The EXIF data for the photo files taken during this test reveal that the manipulated GPS Latitude has been embedded in the photo. Without both the Latitude and Longitude value, it is difficult to determine the GPS location that this photo was taken anyway. However, even with both values, the fact that the manipulated GPS Latitude value has been embedded in the photo suggests that this EXIF GPS data cannot be relied on.

5.5. Conclusion

What is clear from this analysis is that each component in this investigation relies on one another working together to successfully complete a flight. The Parrot AR Drone 2.0 relies on the Flight Recorder to provide accurate GPS data for the userbox file. The same userbox file which is then transferred to the flight controller and is the only source of GPS data available from the flight controller. This is demonstrated clearer in Figure 22. If the EXIF data of the media files stored both the Latitude and Longitude values, then an investigator would not necessarily need to rely on the userbox file transferred from the Parrot AR Drone 2.0.
The Parrot AR Drone 2.0 relies on the flight controller to provide an accurate time and date which is reflected on all of the files found across both of these components. The time and date is reflected on the file name for the video files stored on the Flight Recorder, but as the EXIF data reflects the correct time and date, this suggests that the Flight Recorder isn’t entirely dependent on the flight controller for a time or date. This is demonstrated clearer in Figure 23.
Finally, the Parrot AR Drone 2.0 relies on the Flight Recorder to store video files, and the flight controller to store photo files. This is demonstrated clearer in Figure 24. Photo files can, however, be stored on the Parrot AR Drone 2.0.

Figure 23. Relationship between the Main Components for Time and Date.

Figure 24. Relationship between the Main Components for storing Media files.
In conclusion, this project provides an additional investigation to a limited area of research. The findings of this project can help contribute to future investigations which involve UAVs. As discussed, the Parrot AR Drone 2.0, the flight controller and the Flight Recorder either together, or as separate components, can provide valuable data to an investigation. The main concern highlighted in this study is that the time and date, and the GPS, can be manipulated and therefore should not be definitively relied on for validity. The volatility of data stored on the Parrot AR Drone 2.0 should also be noted as this can have a serious impact on an investigation.
6. Conclusion

This paper presents the first forensic analysis of the Parrot AR Drone 2.0 GPS Edition and other peripheral components including the Flight Recorder and a Samsung Galaxy S4 Mini as the flight controller. The findings and discussion of this project have identified that flight data, specifically GPS data, and photo and video data can be identified from the Parrot AR Drone 2.0 and the flight controller. The only form of data available from the Flight Recorder is video files from which a limited amount of EXIF data can be identified.

As UAV’s continue to increase in popularity, it is expected that the number of UAV-related incidents and offences will also continue to increase. It is hoped that the results and analysis presented in this paper will help contribute to future research and investigations. As discussed in Section 2.6.3, it is expected that the UK will introduce a UAV registration process in 2018. One of the main challenges identified in the research and review of literature presented in Section 1.1 and Section 2.4 is the difficulty in identifying the legitimate owner of a UAV. The introduction of a UAV registration process should contribute to mitigating this challenge. In most of the cases identified in Section 2.4 and in this paper discussed in Chapter 4, the serial number of the UAV can usually be identified on the UAV and flight controller. Should a registration process be introduced, and an investigator is able to identify the serial number of the UAV, they can then identify the owner. However, there is still an issue that will likely remain. In the event of a criminal investigation, an investigator will find difficulty in identifying who actually committed the crime without direct evidence. Furthermore, both previous work discussed in Section 2.4 and the findings presented in Section 4.6 have demonstrated that the manipulation of time and date and GPS data is a simple and straightforward process. Consideration should be made to these findings during an investigation. The validity and reliability of time and date and GPS data should not be taken at face value.

As discussed in Section 1.1, it is expected that the features and components of UAV’s will continue to advance. It is important that research related to both the forensic and security aspect of these future UAV’s will continue.

6.1. Evaluation of Aims and Objectives

The aim of this study is to perform an in-depth forensic analysis of the Parrot AR Drone 2.0 GPS Edition and its peripheral components including the flight controller and Flight Recorder.
The aim of this project was successfully achieved by completing the objectives set before beginning the project. A forensic analysis of the Parrot AR Drone 2.0 was achieved, as well as, a forensic analysis of two other main components. These included the flight controller and the Flight Recorder.

- To review relevant literature related to UAVs and mobile forensics.
This objective was achieved early on in the project. As one of the two main components of this investigation, it was essential that a complete understanding of how UAVs work was understood. An understanding of the current scope of research related to UAV forensics and security was also important. This review helped contribute to achieving the second objective of this project. As the second main component of this investigation, an understanding of mobile phone forensics. Without this, it would have been difficult to decide on the most suitable method of acquisition. Research into the area of Android helped to understand how this OS works.

- To design and implement a suitable practical experiment to gather data.
Review of the current research related to UAV forensics and security helped to achieve this objective. The original scenarios created are inspired by previous work. As a result, the design an experiment was created. However, this design was revisited in Chapter 4 as a result of limitations discovered early in the implementation stages. The revisited methodology provided a suitable method of successfully generating data.

- To analyse the data acquired during testing and identify the areas of importance.
This objective is achieved in Chapter 4 having successfully achieved the second objective. An examination and analysis of the Parrot AR Drone 2.0, the flight controller and the Flight Recorder is done using forensic software. The areas of importance are identified and summarised in three tables specific to each component. This is shown in Section 5.1.

- To discuss the findings of the project and how this could be considered important in the future of forensic investigations.
This objective is achieved in Chapter 5. A comparison to the current research in UAV forensics is discussed. A discussion of how and why the findings of this project can be considered relevant for law enforcement is also made. A correlation between the Parrot AR Drone 2.0, the flight controller and the Flight Recorder is made in Section 5.3. This section discusses whether
the other components involved in a scenario can be identified if an investigator is presented with only one of the main components.

6.2. Future Recommendations

6.2.1. Acquisition of the Parrot AR Drone 2.0

Due to the time limitation and resources available, the method of acquisition for the Parrot AR Drone 2.0 in this investigation is not considered forensically sound. A physical acquisition, specifically Chip-Off, may be better suited for a complete and forensically sound acquisition of this UAV. Additionally, an acquisition in this manner may present further evidence for analysis. Alternatively, a less intrusive yet forensically sound acquisition method should be investigated. This paper has identified that the Parrot AR Drone 2.0 runs a UBIFS. If there is a less intrusive method of acquisition which can work with the UBIFS which can extend to the attempted acquisition made in this paper, then this should be investigated (Refer to Section 4.3.1).

6.2.2. Alternative Flight Controller

The Samsung Galaxy S4 Mini used in this investigation is running Android 4.4.2. At the time of writing this, Android have released Oreo 8.1. Future work should investigate this OS and compare any significant changes between the findings. Also, the Samsung Galaxy S9 and S9+ are the latest releases in Samsung’s Galaxy series. The newest security features of these devices include a fingerprint scanner and intelligent scan, which uses a face and iris recognition system to unlock the device. Future work should investigate how these features, amongst others, present a challenge for investigators and how this could be an issue in UAV forensics. Alternatively, this investigation is limited to an Android device acting as the flight controller. Future work may use another mobile phone OS, such as iOS or Windows, as the flight controller and compare whether the results differ to those presented in this investigation. As discussed in Section 2.3.2, Parrot’s AR Race 2 mobile application is only available on iOS devices. It would be interesting to identify whether data can be retrieved from this application. For example, can the other participants in AR Race 2 be identified? What data can be retrieved from AR Race 2 that is not available on AR.FreeFlight 2.4.15?
6.2.3. Features of the Parrot AR Drone 2.0

As discussed in Section 4.7, the “Control by Map” and “Return Home” features of the Parrot AR Drone 2.0 could not be tested. If the opportunity is available, both of these features should be examined for flight data. For example, in their study Maarse and Sangers (2016) identified artefacts related to each waypoint and the UAV’s home point (Refer to Section 2.4.2). If these features are tested in the future, this would provide an almost complete investigation of the Parrot AR Drone 2.0 available.

6.2.4. UAV Models and Manufacturers.

As concluded, future work is required in order to cover the wide spectrum of UAV’s available on the market to consumers today. There is a limited amount of research on other UAV models and manufacturers. It would be interesting to compare the results presented from this investigation to the features of another UAV, perhaps one which is significantly cheaper or more expensive. UAV’s are advancing significantly. At the time of writing this, Parrot are currently advertising two of their newer “FPV Drones”. The Parrot Bebop 2 Power – Pack FPV is currently retailing for $599.99. It is described as “the latest drone of the Bebop generation and the most advanced of Parrot’s consumer quadcopters”. The Parrot Bebop 2 Power – Pack FPV incorporates an “immersive flight experience” using the Parrot Cockpitglasses 2 which would add another interesting dimension and component into an investigation.
7. Personal Reflection

I had decided on doing a project which involved UAV’s early on as it was an area which can be constantly added to. The aims and objectives of this project were thought through clearly so that the project could be completed successfully.

I had little to no knowledge of UAV’s prior to this project. Therefore, I had a lot of research to do in order to understand how they worked and the current scope of background research related to the forensic and security aspect. The complexity of the UAV ecosystem meant researching mobile phone devices and understanding how Android works. As a result I have developed my knowledge in these two areas in particular.

Several limitations were presented early in the implementation of the design and methodology. I had to adjust my original design plan whilst still ensuring that the aim and objectives could be successfully met. As a result, I have developed my problem-solving skills.

The examination and analysis was an area where I could combine the forensic skills and knowledge I have gained over the last four years at University with the research I had done in the early stages of this project. I was able to identify the key areas of each component in order to acquire data.

Time management was key during this project. Balancing both this project and other modules has been challenging. Personal deadlines were set to ensure that I stayed on track. However, far too much time was spent on trying to understand the UBIFS of the Parrot AR Drone 2.0. This meant that some of my personal deadlines were not met and that the project took slightly longer than anticipated. Also, I now have a basic understanding of Linux commands.

The completion of this final year project has been both challenging and enjoyable. Over the last four years, there have been various assignments and challenges to overcome. I have expanded my knowledge and skills and I am proud to have concluded my three years at University of Derby with this project.
8. References


Appendix A

This Appendix presents a diagram of the Jain et al’s (2017) 12 phase methodology which is applicable to UAV’s. This methodology been discussed in Section 2.4.1 and in Section 3.3.1.
Appendix B

A complete log of the actions taken throughout the investigation process of this project is presented in this Appendix.

Date: 05th March 2018
Time: 12:00
Location: University of Derby – Markeaton Campus
Room: MS317

Parrot AR Drone 2.0

The first forensic workstation does not have a network card so it cannot be used to connect to the Parrot AR Drone 2.0.

A second forensic workstation is introduced to resolve this issue. Kali Linux 2018.1 is installed on Virtual Machine Workstation Player 14 on the second forensic workstation.

Once the Parrot AR Drone 2.0 is turned on, and the motors are initialised, a Wi-Fi hotspot is available. This is “ardrone2_XXXX”. The forensic workstation is connected to this Wi-Fi.

Kali Linux 2018.1 is booted. The command “nmap 192.168.1.1” is run using the Kali Linux Terminal. This reveals that port 21 (FTP) and port 23 (Telnet) are open.

```
root@kali:~# nmap 192.168.1.1
Starting Nmap 7.60 ( https://nmap.org ) at 2018-03-15 13:55 GMT
Nmap scan report for 192.168.1.1
Host is up (1.7s latency).
Not shown: 996 closed ports
PORT STATE SERVICE
21/tcp open  ftp
23/tcp open  telnet
514/tcp filtered shell
5555/tcp open  freely

Nmap done: 1 IP address (1 host up) scanned in 25.51 seconds
root@kali:~#
```

The command “Telnet 192.168.1.1” provides root access to operating system of the Parrot AR Drone 2.0. Using the command “ls”, the directories are listed.
A look at the `/data/video/boxes/` folder revealed flight data from a previous flight. The Reset button of the Parrot AR Drone 2.0 is found under the battery. This button was pressed in an attempt to factory reset the Parrot AR Drone 2.0. However, the flight data still remained. At this point, it is revealed that the reset button did not change data on the Parrot AR Drone 2.0. Therefore, a forensic acquisition of the current state of the Parrot AR Drone 2.0 was required.

**Acquisition of Parrot AR Drone 2.0**

The command “`fdisk -l`” does not reveal any partitions as hoped. The command “`dd if=/data of=~/Documents/`” was run in an attempt to copy the content of the `/data` directory. However, this did not work.

A USB device formatted as FAT32 was inserted into the one USB peripheral of the Parrot AR Drone 2.0. The command “`fdisk -l`” reveals the path of the USB being `/dev/sdb1`.

```
# fdisk -l
```

```
Disk /dev/sda: 208.4 GB, 208354560000 bytes
255 heads, 63 sectors/track, 32035 cylinders
Units = cylinders of 16065 * 512 = 82252800 bytes

Device Boot Start End   Blocks Id System
/dev/sdal *   1   4178  33554432 c    Win95 FAT32 (LBA)
```

The command “`mount /dev/sdb1 /mnt`” is executed. This mounts the USB device at `/mnt` of the Parrot AR Drone 2.0. Access to the content of the USB device is possible if the USB device is mounted and you are connected to the Parrot AR Drone 2.0 via Telnet.

An attempt was made to copy the content of the `/data` to the USB device. The command “`dd if=/data of=/mnt`” is executed. This was unsuccessful.
The USB device is mounted to the Parrot AR Drone 2.0 again.

Further attempts are made to use the “dd” command to copy the entire file system, or individual directories. These attempts are unsuccessful. It is still unclear what filesystem the Parrot AR Drone 2.0 uses.

The command “df –h” is executed revealing the volumes. At this point, it is identified that the Parrot AR Drone 2.0 is running an Unsorted Block Image File System (UBIFS).

The “dd” command was used in an attempt to copy an entire volume. The command is executed as follows “dd =/dev/ubi2 of=/mnt”. This returns the error “/dev/ubi2: Invalid Argument”.

The Narkive Mailing List Archive (2015) provided further information. According to this, read and write access is only implemented for volume devices /dev/ubiX_Y.

Examination of /dev reveal several files named “ubiX_Y”. An attempt using the “dd” command is made. This is executed as “dd if =/dev/ubi2_1 of=/mnt/ParrotData.dd”. This command appears to have completed successfully.
The USB device is unmounted from the Parrot AR Drone 2.0. The USB device is inserted into the first forensic workstation. Some data appears to be in the file.

Autopsy 4.3.0 is launched. A new case is created and the “ParrotData.dd” file is added to the case. This file cannot be opened in Autopsy 4.3.0 as the “file system is unrecognised”.

A new case is created in EnCase v8.05. The “ParrotData.dd” file is added to EnCase. This cannot be opened in EnCase either.

These steps are repeated with “ubi1_0”.

Although hopeful, these attempts were unsuccessful.

The “ParrotData.dd” is copied from the USB device to the ~/Documents.

Date: 13th March 2018
Time: 11:00
Location: University of Derby – Markeaton Campus
Room: MS317

Research was done on the UBIFS, specifically, whether a forensic acquisition of this system could be done and how. It was found that an acquisition may be possible by creating a virtual MTD device.

The command “modprobe nandsim first_id_byte=0x20 second_id_byte=0x71” is executed. This creates a virtual 256MiB MTD device (Linux MTD, n.d.).

Similar work has been done by David Kovar (2015). Kovar (2015) attempted to mount a JFFS2 filesystem. Since UBIFS is designed to overcome JFFS2, the steps suggested are followed. The kernel modules listed at are installed. The MTD device is now shown in /dev.

The instruction is to then copy the image and mount the block device. The command “dd if=~/Documents/ParrotData.dd of=/dev/mtdblock0” is executed. This is done
successfully so now I believe that the “ParrotData.dd” image is copied into “mtdblock0”. An attempt to mount the “mtdblock0” device is done by executing the command “mount -t ubifs /dev/mtdblock0 /mnt”. This is unsuccessful and the following error is reported.

Date: 14th March 2018
Time: 11:00
Location: University of Derby – Markeaton Campus
Room: MS317

The virtual MTD device is deleted once Kali Linux 2018.1 is shut down.

The command “modprobe nandsim first_id_byte=0x20 second_id_byte=0x71” is executed. This creates a virtual 256MiB MTD device (Linux MTD, n.d.).

An attempt to mount virtual MTD device to a UBI volume is done by executing the following command “ubiattach -p /dev/mtd0”. (Wagner, 2011).

The corresponding UBI volume is now shown in /dev.

An image can now be flashed to this volume. The command “ubiupdatevol /dev/ubi0_0 ~/Documents/ParrotData.dd” is executed. This completes successfully.

The command “mount -t ubifs /dev/ubifs /mnt” is executed. The same error is given.
An attempt was made to use FTP to retrieve files. However, this was unsuccessful. The connection between the Parrot AR Drone 2.0 and Kali Linux 2018.1 was closed every time an attempt to list the directories was made.

The only option left is to use the command “cp –v –a” to copy each directory to the mounted USB device. This provides the current state of the AR Drone 2.0.

The Samsung Galaxy S4 Mini is reset to Factory Settings. Cellebrite Phone Detective is used to identify the specific Cellebrite needed for the acquisition process. This is identified as being Cable No: 100.

Cellebrite UFED 4PC is used to take a pre-acquisition of the Samsung Galaxy S4 Mini device. This process takes approximately 13 minutes. The acquisition of the Samsung Galaxy S4 Mini is saved to a Seagate External Hard Drive.
The Cellebrite Memory Card Reader device is used with Cellebrite UFED 4PC to take a pre-acquisition of the Micro SD memory card. This is unsuccessful. The Cellebrite Memory Card Reader device does not seem to communicate with Cellebrite UFED 4PC even though the instructions provided by Cellebrite are followed exactly. Cellebrite UFED 4PC does not get past the “Extraction” page as the “Continue” button is greyed out. Due to this limitation, the Micro SD memory card is taken out of this investigation.

The Samsung Galaxy S4 Mini is turned off.

At this stage, I have a pre-acquisition of the Parrot AR Drone 2.0 (acquired using `cp -v -a` on Kali Linux 2018.1) in its current state and a pre-acquisition of the Samsung Galaxy S4 Mini (acquired using Cellebrite UFED 4PC) after a Factory Reset.

**Date: 31st March 2018**

**Time: 13:00**

The Flight Recorder is connected to the Tableau Forensic USB Bridge Model T8-R2 write-blocker.

Using Guymager on Kali Linux 2018.1, a pre-acquisition of the Flight Recorder is taken. This is saved as a “.dd” file. This file is named “FlightRecBeforeDDCapture”.

A USB device formatted as FAT32 is connected to the forensic workstation. The acquired image is copied to the USB device in order to transfer them from Kali Linux 2018.1 to Windows. At this point, I noticed that the MD5 and SHA1 which was generated on Kali Linux 2018.1 is not the same as the transferred image.

**Date: 1st April 2018**

**Time: 12:00**

The Parrot AR Drone 2.0 and Samsung Galaxy S4 Mini are both put on charge in preparation for flying. Both are charged to 100%.

The Samsung Galaxy S4 Mini is connected to the Internet. I sign in to Google Play using a gmail account set up specifically for University work. A search for “Parrot Free Flight” returns
the current application used for piloting the Parrot AR Drone 2.0. This version is “AR.FreeFlight 2.4.15”. This application was last updated August 22nd 2016. This application has been installed over 1,000,000 times (Google Play, n.d.).

The Samsung Galaxy S4 Mini is switched to Airplane Mode and is then switched off.

An acquisition of the Parrot AR Drone 2.0 (using `cp -v -a /data`) will be taken after each flight.

An acquisition of the Flight Recorder using the Tableau Forensic USB Bridge Model T8-R2 and Guymager on Kali Linux 2018.1 will be taken after each flight.

Without access to Cellebrite UFED 4PC on the second forensic workstation, an acquisition of the Samsung Galaxy S4 Mini can only be taken after all the flight scenarios are complete. In order to preserve the integrity of the data, the Samsung Galaxy S4 Mini will be kept in Airplane Mode and turned off between flights.

**First Flight: Standard Test Flight**

**Location: Millennium Wood, Derby**

A flight was started at approx. 13:40. A GPS Firmware update was done before the Parrot AR Drone 2.0 could take off.

Settings were changed to “Outdoor Hull” so the default settings for an outdoor flight were set. The flight was completed at approx. 14:20.

Note: During this flight, the Parrot AR Drone 2.0 was powered off and the battery was unplugged twice. The first time was at approx. 13:55. The second time was at approx. 14:20.

An acquisition of the Parrot AR Drone 2.0 was taken. The data copied from the `/data` directory was copied to a USB device. This data was then copied from the USB to the Windows machine.

An acquisition of the Flight Recorder was taken after the flight. The acquired image is named “FlightRec_01042018_DD”. This is done in the same way as previously. The acquired image is saved to a USB device. The image is then transferred from the USB device to a Windows machine.
Note: I noticed that the files are different sizes. For example, the “FlightRec_01042018_DD” file is exactly 4GB on Kali Linux 2018.1. However, the file is only 3.68GB on Windows.

At this point, the Parrot AR Drone 2.0 is put on charge.

**Date: 2\textsuperscript{nd} April 2018**

**Time: 14:30**

There is a folder on the Flight Recorder called “DRIVERS”. There are two subfolders. One is called “Windows Drivers”. The second is called “Mac Drivers”. Both folders contain an install program. I attempted to install these, however, they both fail.
Location: Millennium Wood, Derby
The second flight is started at approx. 15:20. I had hoped to test the “Control by Map” and “Return Home” function, however, the install failed. Instead, a standard flight is done and photo and video data is taken.
The flight is finished at approx. 15:30 giving me 10 minutes of flight time. The Parrot AR Drone 2.0 is turned off and the battery is unplugged.

An acquisition of the Parrot AR Drone 2.0 was taken. The data copied from the /data directory was copied to a USB device. This data was then copied from the USB to the Windows machine. Note: I noticed that there was no “flight_diestamp_timestamp” folder or userbox file stored in /data/video/boxes. This suggested that the flight I had just done did not exist. I am unsure why this happened. It is possible that removing the battery so quickly after landing did not give the Parrot AR Drone 2.0 enough time to process the data. However, I will continue with the acquisition as there may be changes to other files.

An acquisition of the Flight Recorder was taken after the flight. The acquired image is named “FlightRec_02042018_DD”. This is done in the same was as previously. The acquired image is saved to a USB device. The image is then transferred from the USB device to a Windows machine.

Date: 4th April 2018
Time: 18:50

**Third Flight: Test for /data/video/boxes folder**

**Location: Millennium Wood, Derby**

The aim of this flight is to determine a possible reason for the Parrot AR Drone 2.0 not creating a folder or file at /data/video/boxes/.

The flight is started at approx. 18:55.

Some photo and video data is taken.

The flight finished at approx. 19:08. The battery was unplugged at 19:15. This should give the Parrot AR Drone 2.0 enough time to process the data.

An acquisition of the Parrot AR Drone 2.0 was taken. The data copied from the /data directory was copied to a USB device. This data was then copied from the USB to the Windows machine.

An acquisition of the Flight Recorder was taken after the flight. The acquired image is named “FlightRec_04042018_DD”. This is done in the same was as previously. The acquired image is saved to a USB device. The image is then transferred from the USB device to a Windows machine.

**Date: 5th April 2018**

**Time: 11:00**

**Fourth Flight: Without Flight Recorder**

**Location: Millennium Wood, Derby**

The aim of this flight is to determine where video is stored if the Flight Recorder is not present during flight. It is also to determine whether the GPS data is affected without the Flight Recorder.

A flight is started at approx. 11:10.

Photo and video is taken.

The flight is finished at approx. 11:20.

The Parrot AR Drone 2.0 and Samsung Galaxy S4 Mini are turned off at approx. 11:25.
An acquisition of the Parrot AR Drone 2.0 was taken. The data copied from the /data directory was copied to a USB device. This data was then copied from the USB to the Windows machine.

As the Flight Recorder was not present in this flight, an acquisition is not done.

**Date: 6th April 2018**
**Time: 11:00**

**Fifth Flight: Counter Forensics – Time and Date Manipulation**

**Location: Millennium Wood, Derby**

The Flight Recorder is included in this flight.

The time on the Samsung Galaxy S4 Mini is changed to 06:00. The date is changed to 2nd January 2010.

The actual flight time of the flight is approx. 18:20.

The manipulated time is 06:08.

Photo and video is taken during this flight.

The actual flight time is finished at approx. 18:35.

The manipulated time is 06:22.

The actual time that the Parrot AR Drone 2.0 and the Samsung Galaxy S4 Mini are turned off is 18:41.

The manipulated time that the Parrot AR Drone 2.0 and the Samsung Galaxy S4 Mini are turned off is 06:27.

An acquisition of the Parrot AR Drone 2.0 was taken. The data copied from the /data directory was copied to a USB device. This data was then copied from the USB to the Windows machine.

An acquisition of the Flight Recorder was taken after the flight. The acquired image is named “FlightRec_TimeDate_DD”. This is done in the same was as previously. The acquired image
is saved to a USB device. The image is then transferred from the USB device to a Windows machine.

Date: 7th April 2018
Time: 16:00

Fifth Flight: Counter Forensics – GPS Manipulation
Location: Millennium Wood, Derby

The Samsung Galaxy S4 Mini is connected to the Internet. The term “GPS Spoof” is typed into Google Play. The application “GPS Emulator” is downloaded. This application will be used to spoof the GPS location. This application was updated on 20th November 2017. This application has been downloaded over 100,000 times (Google Play, n.d.).
A random location is chosen. The coordinates for this location is 43.746013, -79.310140. This places the spoofed GPS location in Toronto, Canada.

The time and date of the Samsung Galaxy S4 mini is set back to normal.

The Samsung Galaxy S4 Mini is restarted. The location is displayed as Victoria Village which is close to the fake GPS.
A flight is started at approx. 17:45.

Photo and video data are taken during this flight.

The flight is finished at approx. 17:55.

The Samsung Galaxy S4 Mini and Parrot AR Drone 2.0 are turned off at approx. 18:00.
An acquisition of the Parrot AR Drone 2.0 was taken. The data copied from the /data directory was copied to a USB device. This data was then copied from the USB to the Windows machine.

An acquisition of the Flight Recorder was taken after the flight. The acquired image is named “FlightRec_GPSSpoof_DD”. This is done in the same was as previously. The acquired image is saved to a USB device. The image is then transferred from the USB device to a Windows machine.

Date: 9th April 2018
Time: 11:00
Location: University of Derby – Markeaton Campus
Room: MS317

A Physical Extraction > ADB is taken of the Samsung using Cellebrite. This is saved to a new folder on the Seagate External HDD.
This process reads blk0_mmcblk0.bin. This takes approx. 12 minutes to complete.

![Image of Extraction completed successfully]

A new case is created in Autopsy for analysis. This is called “Samsung Galaxy S4 Mini – Post Analysis”. The blk0_mmcblk0.bin is added and verified in Autopsy for examination and analysis.
Parrot AR Drone 2.0
Each of the files acquired from the Parrot AR Drone 2.0 are examined in Autopsy 4.3.0.

Config.ini
The config.ini file contains a lot of configuration information for the Parrot. This file updates each time Parrot is flown as the tag `<flying_time>` is updated. The pre-investigation `<flying_time>` is 1747. At the final flight, this `<flying_time>` is 3194. This file also contains `<drone_serial = PS721800BJ5D072214>`. Also `<ssid_single_player = ardrone2_072214>`. Also, `<ardrone_name = My_ARDrone>.

Syslog.bin
This files contains a record of every event executed by the Parrot AR Drone 2.0. The `<USB Vendor>` and `<USB ProductID>` has a record of every USB device plugged into the Parrot AR Drone 2.0.

Profiles Folder:
`/data/custom.configs/profiles` – Footprint from the controller including the name and serial number of the device. The tag `<profile_desc = .Samsung_GT_I9195:b4eca388873de610>.

Sessions Folder:
`/data/custom.configs/sessions` - stores several files as `<config.########.ini>` which appears to store GPS data – I believe that this GPS data is based off the coordinates given by the Samsung. For example, the flight 07-04-2018 is where the GPS data is spoofed has logged the following coordinates `<latitude = 4.374601400000002e+01>` and `<longitude = -7.9310142999999997e+01>.

```plaintext
latitude = 4.374601400000002e+01
longitude = -7.9310142999999997e+01
```
Mapping this in Google Earth puts this in Toronto, Canada. This is a very accurate log of the area where I chose to spoof the GPS data for the Samsung which has logged on the Parrot.
Also, the flight which took place on 05-04-2018 (No Flight Rec), there was no Flight Recorder/GPS Receiver inserted into the Parrot so there is no GPS data logged in the userbox file. However, the corresponding folder includes a config file (“config.0f889176.ini”) with the coordinates \( \text{latitude} = 5.288987699999998e+01 \) and \( \text{longitude} = -1.5095419999999999e+00 \). Mapping this in Google Earth puts this at the top field in Millennium Wood, Derby, which is a pretty accurate approx. of where I was during this flight.
The flight performed on 02-04-2018 did not generate a userbox file as the Parrot was turned off too quickly. However, a config file (“config.2ac7cf43.ini”) was filed with the coordinates <latitude = 5.2890796999999999e+01> and <longitude = -1.5131450000000000e+00>. This puts the Parrot at the bottom field of Millennium Wood, Derby, which is a pretty accurate log of where this flight was done despite there being no userbox file.
**Userbox File:**

`/data/video/boxes` - Flight log information is stored in a folder with the naming convention “flight_date_time”. Within this folder, a userbox file is generated with the following naming convention “userbox_######”. These 10 random digits are apparently Epoch time.

The userbox file is somewhat readable through Autopsy. An investigator may be able to make out the model of the controller, Parrot SSID and serial number, but not much else. A script has been generated by Darklo to make this a more readable format. This script takes a userbox file and converts it to .GPX format which can be opened in Google Earth.

The `/data` folders for 04, 05, 06 and 07 are copied to Kali. The Userbox_to_gpx.rb script is run against each userbox folder. The .gpx file generated is saved to a USB to view later.

Here is an example of the userbox_to_gpx file for flight 04-04-2018 (Standard Flight):

This is a standard output of the userbox_to_gpx file without any specific testing scenarios.
Here is an example of the userbox_to_gpx file for flight 05-04-2018 (No Flight Recorder/GPS Receiver):

Without the Flight Recorder/GPS Receiver, the userbox file does not log any GPS data.
Here is an example of the userbox_to_gpx file for flight 06-04-2018 (Time/Date CFM):
The fake date/time stamp data is reflected on the userbox file, therefore demonstrating that it is pretty easy to manipulate the timestamp data.
Here is an example of the `userbox_to_gpx` file for flight 07-04-2018 (GPS Spoof App):

Despite the Parrot logging the spoofed GPS coordinates of the Samsung (Sessions folder), the Flight Recorder still records the accurate GPS signals which is logged in the userbox file.

---

**Flight_DATE_TIME:**

The most recent flight will overwrite the previous flight so an acquisition is taken after each flight. It was found that if the user disconnects the battery too quickly, the userbox file is not generated as the data is not processed by the Parrot. This was found after a flight taken on 02-04-2018.

**Without Flight Recorder:**

Photo is stored on Samsung and video is stored on Flight Recorder. The Flight Recorder also acts as a GPS receiver. Without the Flight Recorder, the video data stored on the Samsung and photo data stored on the Parrot. Although a userbox file is created, no data is stored within this file. Also, despite using the Flight Recorder for flights after this, photo data continued to store on the Parrot.
Media

Photo (Stored on Parrot):
At no time is a video stored on the Parrot. Photo data can be stored on the Parrot itself however which is what was found when a test without the Flight Recorder was done. With each flight taken from the 04-04-2018, some photo data stored on the Parrot. This photo data was stored in /data/video/boxes/flight_datestamp_timestamp. The naming convention for these files is “picture_date_time.jpg”.

A few images from each test (day) are inserted into Autopsy for analysis.

TIMEDATE TEST: The inaccurate time and date is embedded into the EXIF data of the image. Even viewing this via ExifTool, the time and date is the inaccurate one.

GPS:
04-04-2018: There is no GPS data available via the images. Using an online EXIF tool designed by Friedl (2006), the <Create Date> tag displays the incorrect time, the GPS is almost accurate with somewhat accurate Latitude, however, without a Longitude, this places the image at an inaccurate location.

<table>
<thead>
<tr>
<th>GPS Latitude Ref</th>
<th>North</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS Latitude</td>
<td>52.890833 degrees</td>
</tr>
<tr>
<td>GPS Longitude Ref</td>
<td>West</td>
</tr>
<tr>
<td>GPS Longitude</td>
<td>0.000000 degrees</td>
</tr>
<tr>
<td>GPS Altitude Ref</td>
<td>North</td>
</tr>
<tr>
<td>GPS Altitude</td>
<td>0 m</td>
</tr>
</tbody>
</table>

The same test is done for the 05-04-2018 (No Flight Recorder). We get a somewhat accurate Latitude, however, without the Longitude, it is difficult to determine where this image was taken. As there was no Flight Recorder present during this test, I assume that what GPS information is embedded into the images is taken from the Samsung.

<table>
<thead>
<tr>
<th>GPS Latitude Ref</th>
<th>North</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS Latitude</td>
<td>52.889722 degrees</td>
</tr>
<tr>
<td>GPS Longitude Ref</td>
<td>West</td>
</tr>
<tr>
<td>GPS Longitude</td>
<td>0.000000 degrees</td>
</tr>
<tr>
<td>GPS Altitude Ref</td>
<td>North</td>
</tr>
<tr>
<td>GPS Altitude</td>
<td>0 m</td>
</tr>
</tbody>
</table>
Using this same site, the images taken during 07-04-2018 (GPS SPOOF APPLICATION) are tested.

Again, this displays a somewhat accurate Latitude, however, with the Longitude coordinates, we cannot determine the actual location. Plus, the spoofed Latitude appears to have been embedded into the image, so this is inaccurate anyway.

<table>
<thead>
<tr>
<th>GPS Latitude Ref</th>
<th>North</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS Latitude</td>
<td>43.748833 degrees</td>
</tr>
<tr>
<td>GPS Longitude Ref</td>
<td>West</td>
</tr>
<tr>
<td>GPS Longitude</td>
<td>0.000000 degrees</td>
</tr>
<tr>
<td>GPS Altitude Ref</td>
<td>Above Sea Level</td>
</tr>
<tr>
<td>GPS Altitude</td>
<td>0 m</td>
</tr>
</tbody>
</table>

Counter Forensics

Time/Date:

The Parrot relies on the controller for accurate Time/Date. The “flight_date_time” folder reflects the time/date change made to the Samsung.

GPS:

**Flight Recorder:** The userbox file does not store GPS information if the Flight Recorder is not present. However, the `/data/custom.configs/sessions` stores GPS data according to the GPS given by the Samsung.

**GPS Spoofing Application:** The GPS data stored at `/data/custom.configs/sessions` (GPS via Samsung) can be manipulated using a GPS spoofing application. So, if an investigator is relying solely on this, it may not always be accurate.

SAMSUNG GALAXY S4 Mini

Several .xml files are found at `userdata/data/com.parrot.freeflight`. The following is the naming convention `<MAC Address>_<DateTime>`. They appear to correlate with each flight taken. Each file contains information such as date/time the flight was started `<info key="FLIGHT_TIMESTAMP_FLIGHT_STARTED" value="2018-04-07T17:48:52+0000"/>` and ended `<info key="FLIGHT_TIMESTAMP_FLIGHT_ENDED" value="2018-04-07T17:49:12+0000"/>,` the MAC address of the controller, the controller model and OS
version, the serial number of the Parrot <info key="FLIGHT_DRONE_SERIAL" value="PS721800BJ5D072214"/>

Some files have the time/date that the application started <info key="APPLICATION_SESSION_TIMESTAMP_APPSTARTED" value="2018-04-01T13:07:03+0000"/> and ended <info key="APPLICATION_SESSION_TIMESTAMP_APPLICATION_CLOSED" value="2018-04-01T13:07:40+0000"/>

A .xml file stored as /userdata/data/com.parrot.freeflight/shared_prefs stores what appears to be the last known preference/settings of the user. According to this, the FreeFlight application was opened 12 times.

Counter Forensics

Time/Date:

A .xml file reflecting the fake time/date has been stored. This file contains the same information found in other files (Parrot serial number, flight start/stop, controller model and OS version), however, unlike the other files, this one has not stored the MAC address of the Samsung in its name. The MAC address is not found stored within the file either.

As reference:

<table>
<thead>
<tr>
<th>flight_DATE_TIME</th>
<th>flight_DATE_TIME</th>
<th>flight_DATE_TIME</th>
<th>flight_DATE_TIME</th>
<th>flight_DATE_TIME</th>
</tr>
</thead>
</table>

Media

Photo

A folder appears to have been automatically created at /userdata/media/0/DCIM/AR.Drone where all media related files are kept. There are 23 folders with the naming convention <flight_DATE_TIME>. All of these folders contain a userbox file, the same as the Parrot, some folders also contain photos. Although a userbox file is stored on the Samsung also, most of the data is not readable. Data such as the serial number of the Parrot, the SSID of the Parrot, the make and model of the controller and the date and time of flight is readable, but not ideal.

Following the same convention as earlier, some of the userbox files stored on the Samsung are extracted. In Kali, the userbox_to_gpx.rb file is used to convert these Samsung userbox files to GPX. Upon opening these, they appear to be in the same format as those which were found on the Parrot whereby they log GPS coordinates etc. As a test, a userbox file which correlates between both the Parrot and Samsung is converted. Upon examining this, it is the exact same.

There is no difference, and no different data is found on the Samsung userbox file than that
found on the Parrot userbox file. It would appear that these userbox files are transferred to the controller as the next time that the Parrot is flown, this data is overwritten.

The photos follow the naming convention “picture_datestamp_timestamp”. These store as .jpg files.

There is one video file “video_20180405_111220”. This stores as an .mp4. This is when the test without the Flight Recorder was done.

**Counter Forensics**

**Time/Date:**

The fake time and date is reflected on the photo and embedded into the information when viewed through Autopsy.

**EXIF Data:**

As there are a lot of photos, two photo from each test (day) will be taken and tested against an EXIF tool. The video will also be extracted to determine if any GPS data can be extracted as this was recorded without the Flight Recorder. Some EXIF data can be viewed through Autopsy such as the creation date/time, image width and height, and the make and model of the device which took the photo `<tiff:Make: Parrot AR.Drone>,<tiff:Model: Samsung SOC1040>`. According to Autopsy, the geo latitude coordinates are around 52.890833, latitude -0. The latitude is the same for all media, the longitude varies very slightly on some photos. This is a far from accurate recording.

The GPS data given for 07-04 (GPS SPOOF APP) are 43.745833, -0. This puts the image somewhere in France whereas the spoof GPS was in Toronto, Canada.

The EXIFTool is run against some of these images. It doesn’t appear that this is any different from the information provided by Autopsy. This means that it has no effect on the photos with the fake time/date stamp. The file creation time and date reflect the fake time and date given. The GPS data is not accurate either and gives a similar output to that generated by Autopsy.
Other Relevant Data

Wi-Fi:
The “wpa_supplicant.conf” file stored at /userdata/misc/wifi/ contains information about which Wi-Fi spots that the Samsung has connected to including the Wi-Fi emitted by the Parrot. Therefore, confirming that a connection has been made.

```
network=
    ssid="ar.drone_072214"
    key_mgmt=NONE
    priority=3
    frequency=2412
    autojoin=1
    usable_internet=0
    skip_internet_check=0
```

At /efs/wifi is a file called “.mac.info” which stores the MAC address of the Samsung. MAC Address: B4:3A:28:5B:B7:74

At /efs/FactoryApp there is a file called “serial_no” which contains the serial number of the Samsung. This may be useful in pairing the Parrot AR Drone 2.0 and Samsung Galaxy S4 Mini.

FlightRecorder - PostAnalysis
Files are saved in folders onto the Flight Recorder as “media_datestamp_timestamp”. It appears that only video files save to the Flight Recorder, not photos. Video files are saved as “video_datestamp_timestamp”. These are saved as .mp4 files. Videos can be viewed in Autopsy.

GPS
No GPS data was found to be retrieved from the Flight Recorder.

Counter Forensics

TimeDate:
The changed time/date is reflected on the folders where media is stored.
This changed time/date is also reflected on the saved video files stored within these folders.
With EXIFTool:
A flight from each test (day) is extracted to be run through the EXIFTool application. Obviously, the video file for the 05-04-2018 (No Flight Rec) was examined earlier as this is saved on the phone.

02-01-2010 - (06-04-2018) (TIMEDATE)
Running the TimeDate video through ExifTool reveals the actual creation time and date for the video. Note: The actual start time of this flight was 06-04-2018 at approx. 18:20, so I believe that this is correct. This suggests that, unlike the photo data stored on the Samsung, the video file can get it’s time/date information from somewhere other than the Samsung. In fact, this ExifTool output has no trace of the fake date/time.

| Media Create Date       | : 2018:04:06 18:23:31 |
| Media Modify Date       | : 2018:04:06 18:23:31 |

| Content Create Date     | : 2018:04:06 18:23:31+00:06 |

The other test flight time/date stamps are accurate.